

Fogalmak

Adatbázis (naiv definíció):

1. Egymással logikailag összefüggő, egymáshoz kapcsolódó, belső jelentéssel bíró adatok összessége (kollekciója, együttese).
 2. Az adatbázis speciális célra tervezett, felépített és közzétett adatok együttese.
- Véletlen adatösszesség nem adatbázis. Van egy célcsoport és előre elképzelt alkalmazások egy köre.

Adat: Olyan ismert tény, amely számszerűsíthető és implicit (magától értetődő) jelentése van.

Kisvilág: A valós világ egy része, amelyről az adatbázis az adatokat tárolja.

Adatbázis-kezelő rendszer (DBMS): Olyan szoftvercsomag/rendszer, amely számítógépes adatbázisok létrehozását és karbantartását támogatja.

Adatbázisrendszer: A DBMS szoftver magával az adatokkal együtt. Néha az alkalmazásokat is beleértjük.

Egyed: A valós világnak az az eleme (tárgy, jelenség, elképzelés, személy, fogalom stb.), amely a modellezés tárgyát képezi.

Tulajdonság: Az egyednek a modellezés szempontjából lényeges jellemzője.

Tulajdonságtípus: Az azonos szerepű tulajdonságok absztrakciója.

A tulajdonságtípusok (attribútumok) osztályozása: egyszerű (atomi), összetett; egyértékű, többértékű (lehet tárolt vagy származtatott), NULL-értékű (definiálatlan érték)

A NULL érték előfordulása: nem alkalmazható; nem értelmezett; ismeretlen; létezik, de hiányzik; nem tudjuk, hogy létezik-e.

Egyed típus: Az azonos tulajdonságtípusokkal rendelkező egyedek absztrakciója.

Kapcsolat: A két vagy több egyed típus egyedei között fennálló viszony.

Kapcsolattípus: Két vagy több egyed típus közötti jól meghatározott viszony.

Kapcsolattípusok osztályozása:

- **A kapcsolat foka:** Meghatározza hogy hány egyed típus vesz részt a kapcsolatban.
- **Bináris kapcsolat számossága:** A bináris kapcsolatok számossága meghatározza azon kapcsolat-előfordulások maximális számát, amelyekben egy egyed részt vehet. Pl: A házassági kapcsolat Magyarországon 1:1 számosságú. A tulajdonlási kapcsolat az autó és a személy között 1:N számosságú. A hallgató és tantárgy közötti tantárgyfelvétel kapcsolat M:N számosságú.
- **Kapcsolat szorossága:** Meghatározza, hogy a kapcsolatban részt vevő egyed típusok minden egyedének részt kell-e vennie legalább egy kapcsolat-előfordulásban.
 - Három fajtája: kötelező, félig kötelező, opcionális
 - Pl: 1. A tulajdonlási kapcsolat az autó és a személy között félig kötelező szorosságú (minden autó egyednek részt kell vennie egy kapcsolat-előfordulásban). 2. A tulajdonlási kapcsolat az autó és az autótulajdonos között kötelező szorosságú. 3. könyv és a könyvtártag közötti kölcsönzési viszony opcionális szorosságú.

A Bachman-féle fogalomrendszer:

	absztrakt	konkrét
egyed	egyed típus	egyed-előfordulás
tulajdonság	tulajdonságtípus	tulajdonság-előfordulás
kapcsolat	kapcsolattípus	kapcsolat-előfordulás

Konceptcionális adatmodell (séma): Véges számú tulajdonságtípussal megadott véges számú egyed-típus és a közöttük fennálló véges számú kapcsolattípus összessége.

Adatbázis: Az adatmodell, valamint az egyed-előfordulások, tulajdonság-előfordulások és kapcsolat-előfordulások együttese.

Adatmodell (naiv definíció): Fogalmak egy olyan összessége, amely leírja az adatbázis szerkezetét, azokat a műveleteket amelyekkel ez a szerkezet módosítható és bizonyos kényszereket, melyeket az adatbázisnak ki kell elégítenie.

Adat absztrakció: Egy adatmodellt használunk arra, hogy a tárolási részleteket elrejtjük és csak az adatbázis konceptcionális képét jelenítsük meg a felhasználóknak. A programok az adatmodellre hivatkoznak az adattárolási részletekkel szemben.

Adatmodellek fajtái:

- Konceptcionális (magas szintű, szemantikus). Olyan fogalmakkal dolgozik, amelyek közel vannak ahhoz, ahogy a legtöbb felhasználó gondolkodik az adatokról. (Nevezik egyed vagy objektum alapú adatmodellnek is.)
- Fizikai (alacsony szintű, belső). Olyan fogalmakkal dolgozik, amelyek azt írják le ahogy az adatok eltárolódnak a számítógépben.
- Implementációs (reprezentációs). Olyan fogalmakkal dolgozik, amelyek a fenti két típus között helyezkednek el. A legtöbb DBMS implementáció ezt használja (pl. a relációs modell).

Adatbázis séma: az adatbázis leírása. Az adatbázis szerkezetének, az adattípusoknak és a megszorításoknak a leírását tartalmazza.

Séma diagram: az adatbázis séma (több szempontból való) szemléltető megjelenítése.

Séma konstruktor: a séma vagy a sémán belül egy objektum egy komponense. Pl.: HALLGATÓ.

Adatbázis állapot: az adatbázis egy időpillanatbeli tartalmára hivatkozunk.

Kezdeti adatbázis állapot: az az adatbázis állapot, amelyet kezdetnek betöltünk a rendszerbe.

Érvényes állapot: egy olyan állapot, amely eleget tesz az adatbázis szerkezetének és megszorításainak.

Felhasználók csoportosítása:

- Szereplők a színpadon: Azok, akik valójában használják és ellenőrzik az adatbázis tartalmát, és azok, akik tervezik, fejlesztik és karbantartják az adatbázis alkalmazásokat. (adminisztrátor, tervező, végfelhasználó)
- Dolgozók a színpad mögött: Azok, akik DBMS szoftvereket és kapcsolódó eszközöket terveznek és fejlesztenek, valamint ilyen rendszereket üzemeltetnek.

Logikai adatfüggetlenség: Annak képessége, hogy a konceptcionális séma anélkül változzon meg, hogy a külső sémáknak és a hozzájuk rendelt alkalmazói programoknak meg kellene változni.

Fizikai adatfüggetlenség: Annak képessége, hogy a belső séma anélkül változzon meg, hogy a konceptcionális sémának meg kellene változnia. Pl. a belső séma megváltozhat azáltal, hogy bizonyos fájl szerkezeteket átszervezünk vagy új indexeket hozunk létre az adatbázis hatékonyság javítása miatt.

DDL (Data Definition Language): A DBA (adatbázis adminisztrátor) és adatbázis tervezők használják azért, hogy az adatbázis konceptcionális sémáját meghatározzák.

DML (Data Manipulation Language): Arra használjuk, hogy az adatbázisból való visszakeresést és annak frissítését specifikáljuk.

Reláció: A reláció értékek egy táblázata, amely sorok egy halmazából áll.

Sor: Minden egyes sor adat elemei a modellezett kisvilág egy egyed-előfordulásáról vagy egy kapcsolat-előfordulásáról tartalmaznak tényeket (információt).

Oszlopok: Minden egyes oszlop egy oszlop fejléccel (címmel, attribútum=oszlop fejléc) rendelkezik, amely az illető oszlopban lévő adatok jelentéséről ad információt.

Reláció kulcsa: Minden sor rendelkezik egy olyan adatelem értékkel (vagy azok egy halmazával), amely egyértelműen azonosítja a sort a táblázatban. Ezt kulcsnak nevezzük.

Atomi érték: Atominak nevezzük egy olyan értéket, amely már nem bontható tovább a relációs modell szempontjából.

Tartomány: Egy D tartomány atomi értékek egy halmaza. (Jellemzői: név, adattípus, formátum, korlátozás, további információk az értelmezéshez.)

Relációséma: Relációséma alatt az $R(A_1, A_2, \dots, A_n)$ jelölést értjük, ahol R a relációséma neve, $A_1, A_2, \dots, A_n$ pedig **attribútumok**. Minden A_i attribútum egy szerepkör neve, amelyet valamely D tartomány játszik. D-t az A_i attribútum tartományának nevezzük, és $\text{dom}(A_i)$ -vel jelöljük, n a reláció **foka**.

Pl.: HALLGATÓ(Név, Személyi_száma, Lakcím, Szak, Évfolyam, Neptun_kód)

Reláció: A $R(A_1, A_2, \dots, A_n)$ relációséma egy r relációja – amit szokás $r(R)$ -rel is jelölni – elem n -eseknek egy halmaza: $r = \{t_1, t_2, \dots, t_m\}$.

Minden t_i elem n -es ($1 \leq i \leq m$) n darab értéknek egy rendezett listája: $t_i = \langle v_1, v_2, \dots, v_n \rangle$, ahol minden v_j érték ($1 \leq j \leq n$) vagy $\text{dom}(A_j)$ -nek az eleme, vagy egy speciális NULL érték.

(A definícióban említett elem n -eseket rekordoknak is nevezzük.)

Reláció – másképpen fogalmazva: A relációs adatmodellben egy $r(R)$ reláció nem más, mint egy $\text{dom}(A_1), \text{dom}(A_2), \dots, \text{dom}(A_n)$ tartományokon értelmezett n -ed fokú matematikai reláció, amely részhalmaza azon tartományok Descartes-szorzatának, amelyek R -et definiálják:

$$r(R) \subseteq \text{dom}(A_1) \times \text{dom}(A_2) \times \dots \times \text{dom}(A_n).$$

Ebben a definícióban a NULL értéket beleértjük a A_i attribútumok tartományaiba.

Reláció – jelölések:

- $R(A_1, A_2, \dots, A_n)$ – egy n -ed fokú relációséma
- Q, R, S – relációsémák nevei
- q, r, s – reláció (állapoto)k nevei
- t, u, v – rekordok
- Egy reláció neve a reláció aktuális állapotát jelöli (azaz magát a relációt), míg az $R(A_1, A_2, \dots, A_n)$ alak kizárólag a relációsémára hivatkozik.
- Egy A attribútum minősíthető annak a R relációsémának nevével, amelyhez tartozik: R.A.
- $t = (v_1, v_2, \dots, v_n)$ – egy t elem n -es egy $r(R)$ relációban, ahol v_j az A_j attribútumnak megfelelő érték.
- $t[A_j]$ és $t.A_j$ – az A_j attribútumnak megfelelő v_j érték a t rekordban
- $t[A_u, A_w, \dots, A_z]$ és $t.(A_u, A_w, \dots, A_z)$ – a listában megadott attribútumoknak megfelelő $(v_u, v_w, \dots, v_z)$ értékű részrekord a t rekordból

A relációs modell megszorításai:

- Az adatmodellben benne rejlő megszorítások: modellalapú vagy implicit megszorítások.
- Az adatmodell sémáiban közvetlenül kifejezett megszorítások: sémaalapú vagy explicit megszorítások.
- Olyan megszorítások, amelyeket nem lehet közvetlenül az adatmodell sémáiban kifejezni, és ezért az alkalmazói programokkal kell kifejezni és érvényre juttatni őket: alkalmazásalapú vagy szemantikus megszorítások vagy üzleti szabályok.

Sémaalapú megszorítások:

- tartomány megszorítások
- kulcsmegszorítások és a NULL értékekre vonatkozó megszorítások
- egyedintegritási megszorítások
- hivatkozási integritási megszorítások

Tartománymegszorítás: A tartománymegszorítás kimondja, hogy minden rekordban minden egyes A attribútumhoz tartozó értéknek a $\text{dom}(A)$ tartományból kell származnia, és ezen $\text{dom}(A)$ tartományok minden elemének atomi értéknek kell lennie.

Szuperkulcs: Az R relációsémának létezik egy olyan attribútumhalmaza, amely olyan tulajdonságú, hogy tekintve R bármelyik r relációját, az adott relációban nincs két olyan rekord, amelynek az értékei azonosak lennének ezen attribútumokra vonatkozóan. Az attribútumoknak egy ilyen részhalmazát SK-val jelölve, bármely két különböző t_1 és t_2 rekordot kiválasztva R egy r relációjából: $t_1[SK] \neq t_2[SK]$.

Minden ilyen SK attribútumhalmaz az R relációséma szuperkulcsa.

- **Vagy:** Egy $R = \{A_1, A_2, \dots, A_n\}$ relációséma szuperkulcsa minden olyan $S \subseteq R$ attribútum halmaz, amelyre igaz, hogy bármely R feletti legális r relációban nincs két olyan t_1 és t_2 rekord, amelyekre $t_1[S] = t_2[S]$ teljesül.

Kulcs: A K kulcs egy olyan szuperkulcs, amelyből bármely attribútum eltávolítása azt eredményezi, hogy K már nem lesz szuperkulcs többé. Egy kulcs kielégíti a következő két feltételt:

- Bármilyen relációt tekintve, a reláció két különböző rekordjának nem lehetnek azonosak a kulcsban szereplő attribútumokhoz tartozó értékei.
- Minimális szuperkulcs, azaz egy szuperkulcs, amelyből nem tudunk úgy eltávolítani egyetlen attribútumot sem, hogy az egyediségre vonatkozó feltétel továbbra is fennálljon.

Egy K kulcs egyszerű, ha egyetlen attribútum alkotja, egyébként összetett.

Kulcsjelölt: Egy relációsémának egynél több kulcsa is lehet. Ilyen esetben a kulcsok mindegyikét kulcsjelöltnek hívjuk. A kulcsjelöltek közül egy tetszőlegesen kiválasztott lesz az elsődleges kulcs, a többi másodlagos kulcsoknak nevezzük. Minden relációsémának kell, hogy legyen elsődleges kulcsa.

Elsődleges kulcs: A modellező feladata, hogy a kulcsjelöltek közül kiválasszon egyet a relációséma elsődleges kulcsául. Ez a kulcsjelölt lesz az, amelynek az értékeit a relációkban szereplő rekordok azonosítására fogjuk használni. A kulcsmegszorítás szerint a relációsémának mindig rendelkeznie kell elsődleges kulccsal.

Elsődleges attribútum: Az R relációséma egy attribútumát R egy elsődleges attribútumának nevezzük, ha eleme R valamely kulcsjelöltjének. Egy attribútumot másodlagos (leíró) attribútumnak hívunk, ha nem elsődleges attribútum, azaz nem eleme egyetlen kulcsjelöltnek sem.

Egyedintegritási megszorítás: Az egyedintegritási megszorítás kimondja, hogy egyetlen elsődlegeskulcs-érték sem lehet NULL érték. Ha az elsődleges kulcs összetett, akkor annak egyik komponense sem lehet NULL érték.

Külső kulcs: Egy R_1 relációséma FK-val jelölt attribútumhalmaza külső (idegen) kulcsa R_1 -nek, amely hivatkozik az R_2 relációsémára, ha eleget tesz a következő feltételeknek:

- Az FK-beli attribútumoknak és az R_2 PK-val jelölt elsődleges kulcsattribútumainak páronként azonos a tartománya; ekkor azt mondjuk, hogy az FK attribútumok **hivatkoznak** az R_2 relációsémára.
- Bármely $r_1(R_1)$ aktuális állapotának egy t_1 rekordjában egy FK-beli érték vagy megjelenik egy $r_2(R_2)$ aktuális állapotának valamely t_2 rekordjában PK értékeként, vagy az értéke NULL. Az előbbi esetben $t_1[FK] = t_2[PK]$, ekkor azt mondjuk, hogy a t_1 rekord hivatkozik a t_2 rekordra.

Ha e két feltétel teljesül, egy **hivatkozási integritási megszorítás** áll fenn R_1 -ről R_2 -re vonatkozóan.

Egyéb típusú megszorítások: Szemantikus integritási megszorítások Pl: 1. A dolgozó fizetése nem lehet nagyobb a főnökénél. 2. Egy héten egy dolgozó maximum 56 órát dolgozhat egy projekten. Adatok közti függések: funkcionális függés, többértékű függés. Átmenet-megszorítások vagy dinamikus megszorítások.

Relációs adatbázisséma: Egy S relációs adatbázisséma az $S = \{R_1, R_2, \dots, R_m\}$ relációséma-halmaz, valamint integritási megszorítások – IC-vel jelölt – halmazának az együttese.

Relációs adatbázis(állapot): S egy DB relációs adatbázis(állapot) a olyan $DB = \{r_1, r_2, \dots, r_m\}$ reláció(állapot)k halmaza, ahol minden r_i az R_i séma egy relációja, és minden r_i reláció kielégíti az IC-ben megadott integritási megszorításokat.

Relációs adatbázisséma: Relációsémák egy halmaza és az integritási megszorítások összessége.

Absztrakt lekérdező nyelvek:

- relációalgebra
- relációkalkulus
 - rekordalapú
 - tartományalapú

Relációalgebra: A matematikai halmazelméleten alapuló lekérdező nyelv. A lekérdezés egy kifejezés, amelyben az operátorok relációalgebrai műveletek, az operandusok pedig relációk. A lekérdezés eredménye szintén egy reláció.

Relációkalkulus: Az elsőrendű predikátumkalkulusra épülő lekérdező nyelv. A lekérdezés egy speciális alakú kifejezés, amely egy halmazt definiál. A lekérdezés eredménye az előbb említett halmaz által meghatározott reláció.

A relációalgebra műveletei:

- szelekció (σ)
- projekció (π)
- átnevezés (ρ)
- halmazműveletek
- unió ($\cup$), metszet ($\cap$), különbség ($-$)
- Descartes-szorzat (belső szorzat, $\times$)
- összekapcsolás (join)
 - általános összekapcsolás (theta join)
 - egyenlőség alapú összekapcsolás (equijoin);
 - természetes összekapcsolás (natural join)
 - bal oldali/jobbi oldali/teljes külső összekapcsolás (left/right/full outer join)
 - hányados ($\div$)

Szelekció: Általános alakja: $\sigma_{\langle \text{szelekciós feltétel} \rangle}(R)$

R azt a relációt jelöli, amelyből a $\langle \text{szelekciós feltétel} \rangle$ -nek eleget tevő rekordokat válogatjuk ki. A szelekciós feltétel egy logikai kifejezés, amely logikai operátorokkal összekapcsolt részkifejezésekből épül föl.

A részkifejezések alakja a következők valamelyike lehet:

- $\langle \text{attribútum} \rangle \langle \text{hasonlító op.} \rangle \langle \text{konstans} \rangle$ vagy
- $\langle \text{attribútum} \rangle \langle \text{hasonlító op.} \rangle \langle \text{attribútum} \rangle$,

ahol az $\langle \text{attribútum} \rangle$ az R egy attribútumának neve, a $\langle \text{hasonlító op.} \rangle$ a $\{ =, \neq, <, >, \leq, \geq \}$ operátorok egyike, a $\langle \text{konstans} \rangle$ pedig egy konstans érték az attribútum tartományából.

Egy általános szelekciós feltételben a részkifejezéseket az *és*, a *vagy* és a *nem* logikai operátorokkal kapcsolhatjuk össze.

Pl.: Válasszuk ki azokat a dolgozókat, akik vagy a 4-es osztályon dolgoznak, és többet keresnek 325 000 Ft-nál, vagy az 5-ös osztályon dolgoznak, és többet keresnek 390 000 Ft-nál!

Megoldás: $\sigma(\text{Osz}=4 \text{ AND Fizeset}>325\,000) \text{ OR } (\text{Osz}=5 \text{ AND Fizeset}>390\,000)$ (DOLGOZÓ)

Szelekció tulajdonságai:

- A szelekció unáris művelet.
- Az eredményül kapott reláció foka és sémája megegyezik R fokával, illetve sémájával.
- Az eredményül kapott reláció számossága mindig kisebb vagy egyenlő R számosságánál, azaz bármely f feltétel esetén $|\sigma_f(R)| \leq |R|$
- Két egymásba ágyazott szelekciós művelet végrehajtási sorrendje felcserélhető: $\sigma_{f_1}(\sigma_{f_2}(R)) = \sigma_{f_2}(\sigma_{f_1}(R))$
- Minden többszörösen egymásba ágyazott (kaszkádolt) szelekció átírható egyetlen szelekcióvá, amelynek a feltétele az eredeti feltételek konjunkciója: $\sigma_{f_1}(\sigma_{f_2}(\dots(\sigma_{f_n}(R))\dots)) = \sigma_{f_1 \text{ AND } f_2 \text{ AND } \dots \text{ AND } f_n}(R)$

A projekció tulajdonságai:

- A projekció unáris művelet.
- Az eredményül kapott reláció fokát és sémáját az attribútum listában szereplő attribútumok határozzák meg: az eredmény sémájában az attribútumok sorrendje megegyezik a listában megadott attribútumok sorrendjével. A foksám a listában megadott attribútumok darabszáma lesz.
- Ha az attribútum lista nem tartalmaz kulcs attribútumot, akkor az eredményül kapott reláció számossága kisebb lehet R számosságánál, ugyanis az eredményben nem jelenhetnek meg duplikált rekordok. Ha az attribútum lista R szuperkulcsa, akkor az eredmény számossága megegyezik R számosságával.
- Két egymásba ágyazott projekciós művelet eredménye megegyezik a külső projekció eredményével: $\pi_{lista_1}(\pi_{lista_2}(R)) = \pi_{lista_1}(R)$, ha $lista_2$ tartalmazza a $lista_1$ -beli attribútumokat, egyébként a bal oldal nem értelmezhető.

Projekció: Általános alakja: $\pi_{\langle \text{attribútumlista} \rangle}(R)$, ahol az $\langle \text{attribútumlista} \rangle$ az R reláció lekérdezni kívánt attribútumainak listája.

Pl.: Adjuk meg a dolgozók vezetéknévét, keresztnévét és fizetését!

Megoldás: $\pi_{Vnév,Knév,Fizetés}(DOLGOZÓ)$

Szelekció és projekció:

Adjuk meg az 5-ös osztályon dolgozók vezetéknévét, keresztnévét és fizetését!

Megoldás: $\pi_{Vnév,Knév,Fizetés}(\sigma_{Osz=5}(DOLGOZÓ))$

Átnevezés: Megváltoztathatjuk a relációnk jelölését és átnevezhetjük az attribútumait is értékadás végrehajtásakor: $TEMP \leftarrow \sigma_{Osz=5}(DOLGOZÓ)$ **Általános alakja:** $\rho_S(B_1, B_2, \dots, B_n)(R)$ vagy $\rho_S(R)$ vagy $\rho(B_1, B_2, \dots, B_n)(R)$. A S a reláció jelölésére használt új szimbólum, $B_1, B_2, \dots, B_n$ az új attribútumnevek.

Átnevezés tulajdonságai: Az átnevezés unáris művelet. Az eredményül kapott reláció foka és számossága megegyezik R fokával, illetve számosságával. Az eredményül kapott reláció sémája: a $B_1, B_2, \dots, B_n$ attribútumokkal meghatározott séma lesz, ha megadtuk őket; megegyezik az R sémájával, ha a $B_1, B_2, \dots, B_n$ nem soroltuk fel.

Uniókompatibilitás: Az $R(A_1, A_2, \dots, A_n)$ és $S(B_1, B_2, \dots, B_n)$ relációkat egymással uniókompatibilisnek (típuskompatibilisnek) mondjuk, ha azonos a foksámuk, és $\text{dom}(A_i) = \text{dom}(B_i)$ minden $1 \leq i \leq n$ esetén. (Az uniókompatibilitás tehát azt jelenti, hogy a két relációnak ugyanannyi attribútuma van, és attribútumaik tartományai páronként megegyeznek egymással.)

Unió, metszet, különbség:

- unió: $R \cup S$
- metszet: $R \cap S$
- különbség: $R - S$

A műveletek tulajdonságai:

- Az unió, a metszet és a különbség bináris műveletek.
- Az eredményül kapott reláció sémája – megállapodás szerint – az első (R) reláció sémájával egyezik meg.
- Az unió és a metszet műveletek kommutatívak: $R \cup S = S \cup R$ és $R \cap S = S \cap R$.
- Az unió és a metszet műveletek asszociatívak: $R \cup (S \cap T) = (R \cup S) \cap T$ és $R \cap (S \cup T) = (R \cap S) \cup T$.
- A különbség művelet általában nem kommutatív: $R - S \neq S - R$.

Descartes-szorzat, belső szorzat:

- Általános alakja: $R(A_1, A_2, \dots, A_n) \times S(B_1, B_2, \dots, B_m)$. Két tetszőleges sémájú reláció között elvégezhető bináris művelet.
- Az eredményül kapott Q reláció egy $n + m$ foksámú reláció, melynek sémája: $Q(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m)$
- $|R|$ -rel és $|S|$ -sel jelölve az eredeti két reláció számosságát, az eredményül kapott reláció

számossága: $|R \times S| = |R| \cdot |S|$

- Az eredményül kapott relációban az eredeti két reláció minden rekordjának összes lehetséges kombinációja szerepelni fog.

Általános összekapcsolás:

- Általános alakja: $R \bowtie \langle \text{összekapcsolási feltétel} \rangle S$
- Bináris művelet, operandusai $R(A_1, A_2, \dots, A_n)$ és $S(B_1, B_2, \dots, B_m)$ sémájú relációk.
- Az eredményül kapott Q egy $n+m$ fokszerű reláció, melynek sémája: $Q(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m)$.
- Az eredményül kapott relációban benne lesz az R és az S relációk rekordjainak minden olyan kombinációja, amely kielégíti az összekapcsolási feltételt.

Egyenlőségen alapuló összekapcsolás: Azt az általános összekapcsolási műveletet, amelynek összekapcsolási feltételében csak az egyenlőségjel (=) szerepel összehasonlító műveleti jelként, egyenlőségen alapuló összekapcsolásnak vagy más szóval equijoin műveletnek nevezzük.

Az egyenlőségen alapuló összekapcsolás eredményeként kapott reláció minden rekordjában van legalább egy pár azonos érték.

Természetes összekapcsolás: Az egyenlőségen alapuló összekapcsolás eredményeként kapott relációban – a rekordokban felbukkanó azonos értékpárok miatt – mindig találhatók „felesleges” értékek. A természetes összekapcsolás műveletét az egyenlőségen alapuló összekapcsolás műveletéből származtatjuk oly módon, hogy az ott kapott relációból eltávolítjuk az összekapcsolás alapjául szolgáló, a hozzájuk tartozó értékek egyenlősége miatt felesleges attribútumok egyikét. Az összekapcsolandó két relációban az összekapcsolás alapjául szolgáló attribútumok nevének meg kell egyezniük. **Általános alakja:** $R * S$.

Az eredményül kapott reláció sémája az eredeti két reláció sémájának az attribútumait tartalmazza, ám az összekapcsolás alapjául szolgáló attribútumok közül páronként csak egyet. Az eredményül kapott reláció foka az eredeti két reláció fokszerűségének az összegénél annyival kevesebb, ahány azonos nevű attribútumot tartalmaznak. Az eredményül kapott reláció számossága 0-tól az eredeti relációk számosságainak szorzatáig terjedhet.

A relációalgebrai műveletek teljes halmaza: Bebizonyítható, hogy a relációalgebrai operátorok $\{\sigma, \pi, \cup, -, \times\}$ halmaza teljes halmaz, azaz bármelyik másik relációalgebrai művelet kifejezhető ezen halmazbeli operátorokkal végzett műveletek sorozataként.

Osztás, hányados:

- Általános alakja: $R \div S$.
- Jelöljük Z -vel az R sémáját alkotó attribútumok halmazát, X -szel az S sémáját alkotó attribútumok halmazát! Az osztás művelete akkor hajtható végre, ha $X \subseteq Z$.
- Jelöljük T -vel az eredmény relációt! Legyen $Y = Z - X$! Ekkor Y lesz a T sémáját alkotó attribútumok halmaza.
- A hányados művelet az alábbi műveletek sorozataként fogható fel:
 - $T_1 \leftarrow \pi_Y(R)$
 - $T_2 \leftarrow \pi_Y((S \times T_1) - R)$
 - $T \leftarrow T_1 - T_2$

Pl.: Adjuk meg az összes olyan dolgozó nevét és lakcímét, aki a 'Kutatás' osztályon dolgozik!

Megoldás:

- $KUTATÁS_OSZT \leftarrow \sigma_{\text{Onév}='Kutatás'}(OSZTÁLY)$
- $KUTATÁS_DOLG \leftarrow (KUTATÁS_OSZT \bowtie_{\text{Oszám=Osz}} DOLGOZÓ)$
- $EREDMÉNY \leftarrow \pi_{\text{Vnév,Knév,Lakcím}}(KUTATÁS_DOLG)$
- Egyben felírva: $\pi_{\text{Vnév,Knév,Lakcím}}(\sigma_{\text{Onév}='Kutatás'}(OSZTÁLY \bowtie_{\text{Oszám=Osz}} DOLGOZÓ))$

Pl.: Adjuk meg azon projektek projektszámait, amelyekhez köze van 'Kovács' vezetéknévű dolgozónak, akár a projekten munkálkodó dolgozóként, akár a projektet irányító osztály vezetőjeként!

Megoldás:

- $KOVÁCSOK(Dszsz) \leftarrow \pi_{Dszsz}(\sigma_{Vnév='Kovács'}(DOLGOZÓ))$
- $KOVÁCS_MUNKÁS_PROJ \leftarrow \pi_{Psz}(DOLGOZIK_RAJTA * KOVÁCSOK)$
- $VEZETŐK \leftarrow \pi_{Vnév, Oszám}(\sigma_{Dszsz=Vez_szsz}(DOLGOZÓ \bowtie OSZTÁLY))$
- $KOVÁCS_IR_OSZT(Osz) \leftarrow \pi_{Oszám}(\sigma_{Vnév='Kovács'}(VEZETŐK))$
- $KOVÁCS_IR_PROJ \leftarrow \pi_{Pszám}(KOVÁCS_IR_OSZT * PROJEKT)$
- $EREDMÉNY \leftarrow (KOVÁCS_MUNKÁS_PROJ \cup KOVÁCS_IR_PROJ)$
- Egyben felírva: $\pi_{Psz}(DOLGOZIK_RAJTA \bowtie_{Dszsz=Szzz}(\pi_{Dszsz}(\sigma_{Vnév='Kovács'}(DOLGOZÓ)))) \cup \pi_{Psz}((\pi_{Oszám}(\sigma_{Vnév='Kovács'}(\pi_{Vnév, Oszám}(DOLGOZÓ))) \bowtie_{Szzz=Vez_szsz} OSZTÁLY)) \bowtie_{Oszám=Osz} PROJEKT)$

Relációkalkulusok: A relációs adatmodellnek az elsőrendű predikátumkalkulusra épülő deklaratív, nonprocedurális absztrakt lekérdező nyelvei.

Míg egy relációalgebrai kifejezés azt írja le, hogy hogyan (milyen kiértékelési sorozatot követően) kapjuk meg a lekérdezni kívánt adatokat, a relációkalkulusok kifejezései azt írják le, hogy mit (milyen követelményeknek, feltételeknek eleget tevő adatokat) szeretnénk eredményképpen látni.

Fajtái: rekordalapú relációkalkulus (sorok felett operál), tartományalapú relációkalkulus (oszlopok felett operál)

A kifejezés általános alakja: $\{t_1.A_j, t_2.A_k, \dots, t_n.A_m \mid \text{FELTÉTEL}(t_1, t_2, \dots, t_n, t_{n+1}, t_{n+2}, \dots, t_{n+s})\}$, ahol $t_1, t_2, \dots, t_n, t_{n+1}, t_{n+2}, \dots, t_{n+s}$ **rekordváltozók**, melyek közül $t_1, t_2, \dots, t_n$ **szabadok**, míg $t_{n+2}, \dots, t_{n+s}$ **kötöttek**. Minden egyes A_i annak a relációnak az attribútuma, amelyen az őt minősítő t rekordváltozó felveszi az értékeit, a FELTÉTEL a rekordalapú relációkalkulus logikai formulája.

Az atomi formulák igazságértékei: Az atomi formulák mindegyike – a rekordok tetszőleges kombinációja esetén – rendelkezik egy igazságértékkel: vagy IGAZ, vagy HAMIS értékű.

Formulák rekurzív definíciója: Egy formula egy vagy több atomi formulának az AND, OR és NOT logikai összekötőjelekkel történő összekapcsolásával építhető fel. Minden atomi formula formula. Ha F_1 és F_2 formulák, akkor $(F_1 \text{ AND } F_2)$, $(F_1 \text{ OR } F_2)$, $\text{NOT}(F_1)$ és $\text{NOT}(F_2)$ is formula.

Az egzisztenciális és az univerzális kvantor: Egy rekordváltozót egy kifejezésben szabadnak vagy kötöttnek definiálhatunk a következő szabályok szerint:

- Egy rekordváltozó előfordulása szabad egy F formulában, ha F atomi formula.
- Egy t rekordváltozó előfordulása szabad vagy kötött egy logikai összekötőjellel alkotott formulában – az $(F_1 \text{ AND } F_2)$, $(F_1 \text{ OR } F_2)$, $\text{NOT}(F_1)$ és $\text{NOT}(F_2)$ alakú formulákban – attól függően, hogy szabad vagy kötött F_1 -ben vagy F_2 -ben (ha előfordul bennük).
- Egy t rekordváltozónak minden F -beli szabad előfordulása kötött az $F' = (\exists t)(F)$ vagy $(\forall t)(F)$ alakú F' -ben. A rekordváltozót az F -beli kvantor köti.

Pl.: $F_1: d.\text{Onév} = \text{'Kutatás'}$

$F_2: (\exists t)(d.\text{Oszám} = t.\text{Osz})$

$F_3: (\forall d)(d.\text{Vez_szsz} = \text{'2 551 208 2219'})$

A d rekordváltozó szabad az F_1 és F_2 formulákban, míg a $(\forall)$ kvantor által kötött az F_3 formulában.

A t rekordváltozó a $(\exists)$ kvantor által kötött az F_2 formulában.

Kvantorokat tartalmazó formulák rekurzív definíciója: Ha F egy formula, akkor $(\exists t)(F)$ is az, ahol t egy rekordváltozó. Ha F egy formula, akkor $(\forall t)(F)$ is az, ahol t egy rekordváltozó.

Pl.: Minden kecskeméti projekt esetén adjuk meg a projekt számát, a projektet irányító osztály számát, valamint az osztályvezető vezetéknevét, lakcímét és születési dátumát!

Megoldás: $\{p.\text{Pszám}, p.\text{Osz}, m.\text{Vnév}, m.\text{Szdátum}, m.\text{Lakcím} \mid \text{PROJEKT}(p)$

AND $\text{DOLGOZÓ}(m)$ AND $p.\text{Phelyszín} = \text{'Kecskemét'}$

AND $(\exists d)(\text{OSZTÁLY}(d)$

AND $p.\text{Osz} = d.\text{Oszám}$ AND $d.\text{Vez_szsz} = m.\text{Szzsz})\}$

Pl.: Adjuk meg azon dolgozók vezeték- és keresztnévét, akik az 5-ös osztály által irányított valamelyik projekten dolgoznak!

Megoldás: $\{e.\text{Vnév}, e.\text{Knév} \mid \text{DOLGOZÓ}(e)$

AND (($\exists x$)($\exists w$)(PROJEKT(x) AND DOLGOZIK_RAJTA(w)
AND x.Osz = 5 AND w.Dszsz = e.Szsz AND x.Pszám = w.Psz))))}

Pl.: Adjuk meg azon dolgozók vezeték- és keresztnévét, akik minden olyan projekten dolgoznak, amit az 5-ös osztály irányít!

Megoldás: { e.Vnév, e.Knév | DOLGOZÓ(e) AND (($\forall x$)(NOT (PROJEKT(x))
OR NOT (x.Osz = 5) OR (($\exists w$)(DOLGOZIK_RAJTA(w)
AND w.Dszsz = e.Szsz AND x.Pszám = w.Psz))))}

Biztonságos kifejezések: Egy kifejezést biztonságosnak nevezünk, ha az eredményében szereplő összes érték a kifejezés tartományából való. Egy kifejezés tartománya az összes olyan érték halmaza, amelyek vagy előfordulnak konstansként a kifejezésben, vagy előfordulnak a kifejezésben hivatkozott relációk valamely rekordjában.

Kifejezés a tartományalapú relációkalkulusban:

{ $x_1, x_2, \dots, x_n$ | FELTÉTEL($x_1, x_2, \dots, x_n, x_{n+1}, x_{n+2}, \dots, x_{n+s}$)}, ahol $x_1, x_2, \dots, x_n, x_{n+1}, x_{n+2}, \dots, x_{n+s}$ tartományváltozók, melyek közül $x_1, x_2, \dots, x_n$ szabadok, $x_{n+1}, x_{n+2}, \dots, x_{n+s}$ kötöttek, a FELTÉTEL a tartományalapú relációkalkulus logikai formulája.

Funkcionális függés: A funkcionális függés egy olyan megszorítás, amely az adatbázis két attribútum halmaza között áll fenn. Tegyük fel, hogy a relációs adatbázissémánknak n attribútuma van: $A_1, A_2, \dots, A_n$; és gondoljunk az egész adatbázisunkra úgy, hogy azt egyetlen univerzális $R = \{A_1, A_2, \dots, A_n\}$ relációsémával írjuk le.

- **Definíciója:** Az R két attribútumhalmaza, X és Y között, $X \rightarrow Y$ -nal jelölt funkcionális függés előír egy **megszorítást** azokra a lehetséges rekordokra, amelyek egy R fölötti r relációt alkothatnak. A megszorítás az, hogy bármely két, r -beli t_1 és t_2 rekord esetén, amelyekre $t_1[X] = t_2[X]$ teljesül, teljesülnie kell $t_1[Y] = t_2[Y]$ -nak is.
- **Más szavakkal:** egy R relációsémában X akkor és csak akkor határozza meg funkcionálisan Y -t, ha valahányszor $r(R)$ két rekordja megegyezik az X értékeken, szükségszerűen megegyezik az Y értékeken is.
- **Jelölés:** Funkcionális függések felírásakor a halmazt jelölő nyitó és záró zárójelek, valamint a halmaz elemeit elválasztó vesszők megállapodás szerint elhagyhatók, ha az attribútumokat egybetűs nevekkal azonosítjuk. (Egyelemű halmazok esetén általában egyébként is elhagyjuk a halmazt jelölő zárójeleket.)
Így például: $\{A, B\} \rightarrow \{C\}$ helyett $AB \rightarrow C$, míg $\{A, B, C\} \rightarrow \{D, E\}$ helyett $ABC \rightarrow DE$ írható. Ha X és Y attribútumhalmazokat jelölnek, akkor a funkcionális függések mindkét oldalán alkalmazható az XY egyszerűsítés a két attribútumhalmaz uniójának jelölésére. Így például: $X \cup Y \rightarrow Z$ helyett $XY \rightarrow Z$ írható.

A funkcionális függések tulajdonságai:

- a reflexivitás szabálya: Ha $X \supseteq Y$, akkor $X \rightarrow Y$
- az augmentivitás szabálya: $\{X \rightarrow Y\} \models XZ \rightarrow YZ$
- a tranzitivitás szabálya: $\{X \rightarrow Y, Y \rightarrow Z\} \models X \rightarrow Z$
- a dekompozíció szabálya: $\{X \rightarrow YZ\} \models X \rightarrow Y$
- az additivitás szabálya: $\{X \rightarrow Y, X \rightarrow Z\} \models X \rightarrow YZ$
- a pszeudotranzitivitás szabálya: $\{X \rightarrow Y, WY \rightarrow Z\} \models WX \rightarrow Z$

Triviális függés: Egy $X \rightarrow Y$ funkcionális függés triviális, ha $X \supseteq Y$, egyébként nemtriviális.

A funkcionális függések tulajdonságai:

- A reflexivitás szabálya szerint egy attribútumhalmaz mindig meghatározza önmagát, vagy saját maga bármilyen részhalmazát.
- Az augmentivitás szabálya szerint egy funkcionális függés mindkét oldalának ugyanazzal az attribútumhalmazzal történő bővítése újabb érvényes funkcionális függést eredményez.
- A tranzitivitás szabálya szerint a funkcionális függések tranzitívak.

- A dekompozíció szabálya azt mondja, hogy egy funkcionális függés jobb oldaláról eltávolíthatunk attribútumokat.
- Az additivitás szabálya szerint funkcionális függések egy $\{X \rightarrow A_1, X \rightarrow A_2, \dots, X \rightarrow A_n\}$ halmazát összevonhatjuk egyetlen $X \rightarrow \{A_1, A_2, \dots, A_n\}$ funkcionális függéssé.

A reflexivitás bizonyítása: Tegyük fel, hogy $X \supseteq Y$, és hogy léteznek t_1 és t_2 rekordok R valamely r relációjában úgy, hogy $t_1[X] = t_2[X]$. Ekkor $t_1[Y] = t_2[Y]$, mivel $X \supseteq Y$; ezért $X \rightarrow Y$ -nak teljesülnie kell r -ben.

Az augmentívitas bizonyítása (indirekt): Tegyük fel, hogy $X \rightarrow Y$ fennáll R egy r relációjában, de $XZ \rightarrow YZ$ nem áll fenn. Ekkor léteznie kell t_1 és t_2 rekordoknak úgy, hogy ① $t_1[X] = t_2[X]$, ② $t_1[Y] = t_2[Y]$, ③ $t_1[XZ] = t_2[XZ]$ és ④ $t_1[YZ] \neq t_2[YZ]$. Ez nem lehetséges, mert ③-ból kapjuk, hogy ⑤ $t_1[Z] = t_2[Z]$, míg ②-ből és ⑤-ből kapjuk, hogy $t_1[YZ] = t_2[YZ]$, ami ellentmond ④-nek.

A tranzitivitás bizonyítása: Tegyük fel, hogy ① $X \rightarrow Y$ és ② $Y \rightarrow Z$ fennáll egy r relációban. Ekkor tetszőleges t_1 és t_2 r -beli rekordokra, melyekre igaz, hogy $t_1[X] = t_2[X]$, ① miatt kapjuk, hogy ③ $t_1[Y] = t_2[Y]$; így ③-ból és a ②-es feltevésünkből azt is kapnunk kell, hogy ④ $t_1[Z] = t_2[Z]$, ezért ezért $X \rightarrow Z$ -nek fenn kell állnia r -ben.

A dekompozíció bizonyítása: ① $X \rightarrow YZ$ adott. ② $YZ \rightarrow Y$, felhasználva a reflexivitás szabályát, és tudva, hogy $YZ \supseteq Y$. ③ $X \rightarrow Y$, alkalmazva a tranzitivitás szabályát ①-re és ②-re.

Az additivitás bizonyítása: ① $X \rightarrow Y$ adott. ② $X \rightarrow Z$ adott. ③ $X \rightarrow XY$, alkalmazva az augmentívitas szabályát ①-re, azt X -szel bővítve; megjegyezve, hogy $XX = X$. ④ $XY \rightarrow YZ$, alkalmazva az augmentívitas szabályát ②-re, azt Y -nal bővítve. ⑤ $X \rightarrow YZ$, alkalmazva a tranzitivitás szabályát ③-ra és ④-re.

A pszeudotranzitivitás bizonyítása: ① $X \rightarrow Y$ adott. ② $WY \rightarrow Z$ adott. ③ $WX \rightarrow WY$, alkalmazva az augmentívitas szabályát ①-re, azt W -vel bővítve. ④ $WX \rightarrow Z$, alkalmazva a tranzitivitás szabályát ③-ra és ②-re.

Armstrong-axiómák: A reflexivitás, az augmentívitas és a tranzitivitás szabályait együtt Armstrong-axiómáknak nevezzük.

Lezárt: Minden egyes ilyen X attribútumhalmazra meghatározzuk az attribútumoknak egy olyan X^+ halmazát, amelyet X funkcionálisan meghatároz F alapján; X^+ -ot X F alatti lezártjának nevezzük.

Funkcionális függések halmazainak lefedése: Azt mondjuk, hogy a funkcionális függések F halmaza lefedi a funkcionális függések egy másik, E halmazát, ha minden E -beli funkcionális függés benne van F^+ -ban; azaz ha minden E -beli függés levezethető F -ból.

Funkcionális függések halmazainak ekvivalenciája: A funkcionális függések E és F halmaza ekvivalens egymással, ha $E^+ = F^+$. Így az ekvivalencia azt jelenti, hogy minden E -beli funkcionális függés levezethető F -ból, és minden F -beli funkcionális függés levezethető E -ből; azaz E ekvivalens F -fel, ha E lefedi F -et és F lefedi E -t.

Funkcionális függések minimális halmaza Funkcionális függések egy F halmazát minimálisnak nevezzük, ha

- Minden $X \rightarrow Y$ funkcionális függésre F -ben Y egyszerű, azaz egy attribútumból áll.
- Nem hagyhatunk el egyetlen funkcionális függést sem F -ból úgy, hogy F -fel ekvivalens halmazt kapjunk.
- Nem helyettesíthetünk egyetlen egy $X \rightarrow A$ funkcionális függést F -ben egy $Y \rightarrow A$ funkcionális függéssel, ahol $Y \subset X$, úgy, hogy F -fel ekvivalens halmazt kapjunk.

Minimális lefedés: Funkcionális függések egy E halmazának minimális lefedése alatt (standard kanonikus alakban lévő redundancia mentes) funkcionális függések egy olyan halmazát értjük, amely minimális és ekvivalens E -vel.

Minimális halmaz jellemzői:

- Funkcionális függések minden halmazának van vele ekvivalens minimális halmaza.
- Több, egymással is ekvivalens minimális halmaz létezhet.

- A minimális halmaz (nem egyértelmű) standard kanonikus alak, redundanciák nélkül.
- Amikor attribútumok egy halmazából relációkat állítunk elő a tervezés folyamán, akkor feltételezzük, hogy funkcionális függések egy minimális halmazából indulunk ki. Pontosabban, először előállítjuk ezt a minimális halmazt a szemantika által adott funkcionális függésekből.

A reláció attribútumainak szemantikája (1. (nem hivatalos) irányelv): Egy reláció minden egyes rekordja egy egyedet vagy kapcsolat-előfordulást reprezentáljon. (Az egyes relációkra és azok attribútumaira külön-külön vonatkozik.) Olyan sémát kell tervezni, ami könnyen magyarázható relációról relációra. Az attribútumok szemantikájának könnyen értelmezhetőnek kell lennie.

Redundáns információk a rekordokban és a karbantartási anomáliák (2. (nem hivatalos) irányelv): Olyan sémát tervezzünk, amelyben nem jelennek meg beszúrási, törlési és módosítási anomáliák. Ha mégis előfordulnak anomáliák, akkor jegyezzük fel azokat, hogy az alkalmazások számításba vehessék őket.

NULL értékek a rekordokban (3. (nem hivatalos) irányelv): A relációkat úgy kell megtervezni, hogy a rekordjaik a lehető legkevesebb NULL értéket tartalmazzák. Azok az attribútumok, amelyek gyakran vesznek fel NULL értéket, külön relációkba tehetők (az elsődleges kulccsal).

Álrekordok (4. (nem hivatalos) irányelv): A relációkat úgy kell megtervezni, hogy kielégítsék a veszteségmentes összekapcsolás feltételét. Egy tetszőleges relációkon végrehajtott természetes összekapcsolás nem állíthat elő álrekordokat.

Normalizáció: A normalizáció az a folyamat, amelynek során szétbontjuk a nem kielégítő, „rossz” relációsémákat úgy, hogy az attribútumaikat több kisebb relációsémába helyezzük át.

Normálforma: A normálforma a relációsémák kulcsai és a bennük fennálló funkcionális függések segítségével megfogalmazott feltétel, amellyel megállapítható, hogy a relációséma egy adott normálformában van-e.

Relációsémák normálformái:

- 2NF, 3NF, BCNF: a relációsémák kulcsai és a bennük fennálló funkcionális függések alapján
- 4NF: kulcsok és többértékű függések alapján
- 5NF: kulcsok és join függések alapján
- További tulajdonságok lehetnek szükségesek a jó relációs tervezés biztosításához (veszteségmentes összekapcsolás, függésmegőrzés)

Első normálforma: Tiltja: Az összetett attribútumokat, a többértékű attribútumokat, a beágyazott relációkat (az olyan attribútumokat, amelyek értékei a különálló rekordokban nem atomiak). A reláció definíciójának részét képezi.

Teljes funkcionális függés: Egy $X \rightarrow Y$ funkcionális függés teljes funkcionális függés, ha X-ből bármely A attribútumot eltávolítva a függés a továbbiakban már nem áll fenn, azaz bármely $A \in X$ attribútum esetén $(X - \{A\})$ már nem határozza meg funkcionálisan Y-t.

Részleges függés: Egy $X \rightarrow Y$ funkcionális függés részleges függés, ha valamely $A \in X$ attribútum eltávolítható X-ből úgy, hogy a függés továbbra is fennáll, azaz valamely $A \in X$ esetén $(X - \{A\}) \rightarrow Y$.

Második normálforma: Egy R relációséma második normálformában (2NF-ben) van, ha R minden másodlagos (leíró) attribútuma teljesen funkcionálisan függ R elsődleges kulcsától.

Egy R relációséma második normálformában (2NF-ben) van, ha R-nek nincs olyan másodlagos (leíró) attribútuma, amely részlegesen függne R bármely kulcsától.

Tranzitív függés: Egy R relációséma $X \rightarrow Y$ funkcionális függése tranzitív függés, ha létezik egy olyan Z attribútumhalmaz, amely nem kulcsjelölt és nem része R egyetlen kulcsának sem, és fennáll $X \rightarrow Z$, illetve $Z \rightarrow Y$.

Harmadik normálforma: Egy R relációséma harmadik normálformában (3NF-ben) van, ha 2NF-ben van, és nincs R-nek olyan másodlagos (leíró) attribútuma, amely tranzitívan függne az elsődleges kulcstól.

Egy R relációséma harmadik normálformában (3NF-ben) van, ha valahányszor egy $X \rightarrow A$ nemtriviális funkcionális függés fennáll R-en, akkor vagy ① X egy szuperkulcsa R-nek, vagy ② A egy elsődleges attribútuma R-nek.

Boyce–Codd-féle normálforma: Egy R relációséma Boyce–Codd-féle normálformában (BCNF-ben) van, ha valahányszor egy $X \rightarrow A$ nemtriviális funkcionális függés fennáll R-en, akkor X egy szuperkulcsa R-nek.

Pl.: Olyan reláció, amely 3NF-ben van viszont nincs BCNF-ben. Legyen R(A, B, C) reláció AB kulccsal és FD1: $AB \rightarrow C$, FD2 : $C \rightarrow B$ funkcionális függésekkel. Ekkor R 3NF-ben van, hiszen AB szuperkulcs (FD1) és B elsődleges attribútum (FD2), viszont nincs BCNF-ben mivel C nem szuperkulcs.

Többértékű függés (MVD): Informálisan : valahányszor két független 1:N számosságú A:B és A:C kapcsolattípust összemixelünk, összemossunk egy R(A, B, C) relációsémába, többértékű függés keletkezhet.

Egy R relációsémán megadott $X \twoheadrightarrow Y$ többértékű függés, ahol X és Y R attribútumhalmazai, a következő megszorítást jelenti bármely R fölötti r reláció esetén: Ha van két olyan t_1 és t_2 rekord r-ben, amelyre $t_1[X] = t_2[X]$, akkor léteznie kell két t_3 és t_4 rekordnak is r-ben a következő tulajdonságokkal, ahol Z-t az $(R - (X \cup Y))$ jelölésére használjuk:

- $t_3[X] = t_4[X] = t_1[X] = t_2[X]$
- $t_3[Y] = t_1[Y]$ és $t_4[Y] = t_2[Y]$
- $t_3[Z] = t_2[Z]$ és $t_4[Z] = t_1[Z]$

Valahányszor $X \twoheadrightarrow Y$, azt mondjuk, hogy X **többértékűen meghatározza** Y-t.

Funkcionális és többértékű függések tulajdonságai:

- A funkcionális függések reflexivitási szabálya: Ha $X \supseteq Y$, akkor $X \rightarrow Y$
- A funkcionális függések augmentitási szabálya: $\{X \rightarrow Y\} \models XZ \rightarrow YZ$
- A funkcionális függések tranzitivitási szabálya: $\{X \rightarrow Y, Y \rightarrow Z\} \models X \rightarrow Z$
- A többértékű függések komplementer szabálya: $\{X \twoheadrightarrow Y\} \models \{X \twoheadrightarrow (R - (X \cup Y))\}$
- A többértékű függések augmentitási szabálya: Ha $X \twoheadrightarrow Y$ és $W \supseteq Z$, akkor $XW \twoheadrightarrow YZ$
- A többértékű függések tranzitivitási szabálya: $\{X \twoheadrightarrow Y, Y \twoheadrightarrow Z\} \models X \twoheadrightarrow (Z - Y)$
- A funkcionális függésből következik a többértékű függés: $\{X \rightarrow Y\} \models X \twoheadrightarrow Y$
- A többértékű függésből bizonyos esetekben következik valamiféle funkcionális függés: Ha $X \twoheadrightarrow Y$ és létezik olyan W, amelyre ① $W \cap Y$ üres, ② $W \rightarrow Z$, és ③ $Y \supseteq Z$, akkor $X \rightarrow Z$

Triviális többértékű függés: Egy $X \twoheadrightarrow Y$ többértékű függést triviális többértékű függésnek nevezünk, ha vagy ① Y részhalmaza X-nek, vagy ② $X \cup Y = R$. Egy olyan többértékű függést, amely sem ①-t, sem ②-t nem elégíti ki, nemtriviális többértékű függésnek nevezünk.

- Egy triviális többértékű függés fenn fog állni R bármilyen r relációján; azért nevezzük triviálisnak, mert nem határoz meg semmilyen lényeges vagy értelmes (jelentéssel bíró) megszorítást R-re vonatkozóan.

Negyedik normálforma: Egy R relációséma negyedik normálformában (4NF-ben) van, figyelembe véve az F függések halmazát (amely magában foglalja a funkcionális és többértékű függéseket), ha minden F^+ -beli nemtriviális $X \twoheadrightarrow Y$ többértékű függés esetén X szuperkulcsa R-nek.

A dekompozíciók veszteségmentes join tulajdonsága: Az R relációséma egy $D = \{R_1, R_2, \dots, R_m\}$ dekompozíciója veszteségmentes join tulajdonságú, figyelembe véve az R-beli F függések halmazát, ha R minden r relációjára, amely kielégíti F-et, fennáll a következő:

$*(\pi_{R_1}(r), \dots, \pi_{R_m}(r)) = r$, ahol * a természetes összekapcsolást jelöli.

A bináris dekompozíciók veszteségmentes join tulajdonságának tesztelése: Az R_1 és R_2 relációsémák akkor és csak akkor alkotják az R egy veszteségmentes join dekompozícióját, figyelembe véve a funkcionális és többértékű függések F halmazát, ha $(R_1 \cap R_2) \rightarrow\rightarrow 1 - R_2$, vagy – szimmetrikusan – akkor és csak akkor, ha $(R_1 \cap R_2) \rightarrow\rightarrow 2 - R_1$.

Kapcsolásfüggés: Egy R relációsémán megadott kapcsolásfüggés (JD) meghatároz egy megszorítást az R bármely r relációjára. A megszorítás azt írja elő, hogy R minden legális r relációjának kell, hogy legyen egy veszteségmentes join dekompozíciója az $R_1, R_2, \dots, R_n$ sémákba; azaz minden ilyen r -re $*(\pi_{R_1}(r), \pi_{R_2}(r), \dots, \pi_{R_n}(r)) = r$. Az így előírt megszorítást $JD(R_1, R_2, \dots, R_n)$ -nel jelöljük.

Triviális kapcsolásfüggés: Egy R sémára megadott $JD(R_1, R_2, \dots, R_n)$ kapcsolásfüggés triviális kapcsolásfüggés, ha valamely $JD(R_1, R_2, \dots, R_n)$ -beli R_i relációséma egyenlő R -rel.

Ötödik normálforma: Egy R relációséma ötödik normálformában (5NF-ben) van, figyelembe véve funkcionális, többértékű és kapcsolásfüggések egy F halmazát, ha minden F^+ -beli nemtriviális $JD(R_1, R_2, \dots, R_n)$ esetén minden R_i szuperkulcsa R -nek.

Gyenge egyedtypus: Azokat az egyedtypusokat, amelyek nem rendelkeznek saját kulcsattribútumokkal, gyenge egyedtypusoknak nevezzük. Ezzel ellentétben azokat a (hagyományos) egyedtypusokat, amelyeknek van kulcsattribútumuk, erős egyedtypusoknak nevezzük.

Részleges kulcs: A gyenge egyedtypusoknak részleges kulcsuk (diszkriminátoruk) van, amely azon attribútumok halmaza, amelyek egyértelműen azonosítják azokat a gyenge egyedeket, amelyek ugyanazon tulajdonos egyed(ek)hez kapcsolódnak.

ER séma leképezése relációs sémára:

- Erős egyedtypusok leképezése
- Gyenge egyedtypusok leképezése
- Bináris 1:1 számosságú kapcsolattípusok leképezése
 - külső kulcs használata
 - összevonás
 - kereszthivatkozás v. kapcsoló reláció használata
- Bináris 1:N számosságú kapcsolattípusok leképezése
- Bináris M:N számosságú kapcsolattípusok leképezése
- Többértékű attribútumok leképezése
- N-edfokú kapcsolattípusok leképezése

Osztály: Egy osztály egyedek egy halmaza vagy kollekciója; magában foglal minden olyan az EER sémabeli szerkezetet, amely egyedeket csoportosít, például egyedtypusokat, alosztályokat, szuperosztályokat és kategóriákat.

Alosztály: Egy S alosztály egy olyan osztály, amely egyedeinek mindig egy másik osztály, a szuperosztály/alosztály (vagy IS-A) kapcsolat C szuperosztályához tartozó egyedek egy részhalmazát kell alkotniuk.

- Egy ilyen kapcsolatot C/S -sel jelölünk. Egy szuperosztály/alosztály kapcsolatra mindig igaz, hogy $S \subseteq C$.

Specializáció: Egy $Z = \{S_1, S_2, \dots, S_n\}$ specializáció olyan alosztályoknak egy halmaza, amelyeknek ugyanaz a G a szuperosztálya, azaz $i = 1, 2, \dots, n$ esetén G/S_i egy szuperosztály/alosztály kapcsolat.

- G -t generalizált egyedtypusnak (vagy a specializáció szuperosztályának, olykor pedig az $\{S_1, S_2, \dots, S_n\}$ alosztályok generalizációjának) nevezzük.

Totális és részleges specializáció: Z -t totálisnak nevezzük, ha mindig (bármely időpillanatban)

$$\bigcup_{i=1}^n S_i = G$$

teljesül, hogy $i = 1$. Egyébként Z -t részlegesnek (parciálisnak) mondjuk.

Diszjunkt és átfedő specializáció: Z-t diszjunktnek nevezzük, ha minden $i \neq j$ esetén teljesül, hogy $S_i \cap S_j = \emptyset$ (üres halmaz). Ellenkező esetben Z-t átfedőnek mondjuk.

Predikátumdefiniált és felhasználó által definiált specializáció: C-nek egy S alosztályát predikátumdefiniáltak nevezzük, ha egy p predikátumot írunk elő a C attribútumaira, amellyel megadjuk, hogy mely C-beli egyedek elemei S-nek; azaz $S = C[p]$, ahol $C[p]$ azon C-beli egyedek halmaza, amelyek eleget tesznek a p feltételnek. Egy alosztályt, amit nem predikátummal definiálunk, felhasználó által definiáltak nevezünk.

Attribútumdefiniált specializáció: Egy Z specializációt (vagy egy G generalizációt) attribútumdefiniáltak nevezünk, ha egy $(A = c_i)$ predikátumot használhatunk minden egyes Z-beli S_i alosztály tagságának a megadására, ahol A G-nek egy attribútuma, c_i pedig egy konstans érték A tartományából.

- Ha $i \neq j$ esetén $c_i \neq c_j$, és A egy egyértékű attribútum, akkor a specializáció diszjunkt lesz.

Kategória: Egy T kategória egy osztály, amely n definiáló szuperosztály ($D_1, D_2, \dots, D_n, n > 1$) uniójának egy részhalmaza. Formálisan: $T \subseteq (D_1 \cup D_2 \cup \dots \cup D_n)$. Egy D_i attribútumaira előírt p_i predikátumot használunk az egyes D_i -k azon elemeinek a megadására, amelyek elemei T-nek. Ha minden D_i -re megadjuk egy p_i predikátumot, akkor $T = (D_1[p_1] \cup D_2[p_2] \cup \dots \cup D_n[p_n])$.

- Ezek után kiterjeszthetjük a kapcsolattípus definícióját, megengedve, hogy bármilyen osztály – ne csak az egyed típusok – részt vehessen egy kapcsolattípusban. Csak ki kell cserélnünk az egyed típus szavakat az osztály szóra a definícióban. Az EER grafikus jelölései konzisztensek az ER-rel, mert az osztályokat is téglalapokkal reprezentáljuk.

EER séma leképezése relációs sémára:

- Erős egyed típusok leképezése
- Gyenge egyed típusok leképezése
- Bináris 1:1 számosságú kapcsolattípusok leképezése
 - külső kulcs használata
 - összevonás
 - kereszthivatkozás v. kapcsoló reláció használata
- Bináris 1:N számosságú kapcsolattípusok leképezése
- Bináris M:N számosságú kapcsolattípusok leképezése
- Többértékű attribútumok leképezése
- N-edfokú kapcsolattípusok leképezése
- Specializációk és generalizációk leképezése
- Unió típusok (kategóriák) leképezése

Specializációk és generalizációk leképezése: Konvertáljunk át minden C (generalizált) szuperosztállyal és m darab, $\{S_1, S_2, \dots, S_m\}$ alosztállyal rendelkező specializációt, ahol C attribútumai $\{k, a_1, \dots, a_n\}$ és k az (elsődleges) kulcs, a következő lehetőségek valamelyike szerint relációsémákká:

- Több reláció – szuperosztály és alosztályok
- Több reláció – csak alosztály relációk
- Egyetlen reláció egy típus attribútummal
- Egyetlen reláció több típus attribútummal

Unió típusok (kategóriák) leképezése: Különböző kulcsokkal rendelkező szuperosztályok által definiált kategória leképezéséhez célszerű egy új kulcsattribútumot bevezetni, amelyet helyettesítő kulcsnak nevezünk a kategóriának megfelelő reláció létrehozásakor.

A helyettesítő kulcs attribútumot minden olyan relációba is felvesszük, amelyeket a kategória szuperosztályaiból képezünk.

Tranzakció (heurisztikus): Adatbázis folyamatok egy olyan logikai egysége, amely egy vagy több adatbázis-hozzáférési műveletet (olvasás-kinyerés, írás-beszúrás, frissítés, törlés) tartalmaz.

Tranzakció: A tranzakció egy végrehajtás alatt álló program, amely az adatbázis-feldolgozás egy logikai egységét alkotja. Egy tranzakció egy vagy több adatbázis-hozzáférési műveletből (beszúrás, törlés, módosítás és lekérdezés) áll.

A tranzakció állapotai és műveletei:

- **Állapotok:** aktív állapot, részlegesen véglegesített állapot, véglegesített (commit) állapot, hibás állapot, megszakított állapot
- **Műveletek:** begin_transaction, read vagy write, end_transaction, commit_transaction, rollback vagy abort, undo (visszaállítás során használható, hasonló a rollback-hez azt kivéve, hogy elemibb művelet minthogy egy egész tranzakcióra vonatkozna), redo (visszaállítás során használható, bizonyos tranzakció műveleteket újra végrehajt a biztonság kedvéért)

Helyreállítás a rendszer loggal: Ha a rendszer összeomlik, akkor helyreállíthatjuk konzisztens adatbázis állapotba a log vizsgálatával megfelelő módszerek révén.

Véglegesítési (commit) pont: Egy tranzakció akkor éri el a véglegesítési (commit) pontját, ha az összes adatbázis-hozzáférési művelete sikeresen végrehajtott és ezen műveletek hatása kiírásra került a log-fájlba.

ACID tulajdonságok:

- **Atomosság** (atomicity): A tranzakció a feldolgozás atomi egysége; vagy teljes egészében végrehajtodik, vagy egyáltalán nem.
- **Konzisztenciamegőrzés** (consistency preservation): Egy tranzakció konzisztenciamegőrző, ha teljes és önálló végrehajtása az adatbázist konzisztens állapotból konzisztens állapotba viszi át.
- **Elkülönítés** (isolation): Egy tranzakciónak látszólag más tranzakcióktól elkülönítve kell végrehajtodnia. Ez azt jelenti, hogy a tranzakció végrehajtása nem állhat kölcsönhatásban semelyik másik konkurensen végrehajtott tranzakcióval sem.
- **Tartósság vagy állandóság** (durability vagy permanency): Egy véglegesített tranzakció által az adatbázison véghezvitt módosításoknak meg kell őrződniük az adatbázisban. Ezeknek a módosításoknak semmilyen hiba miatt nem szabad elveszniük.

Az ACID tulajdonságok biztosításáért a DBMS konkurenciavezérlő és naplózó/helyreállító alrendszerei a felelősek.

Tranzakció ütemezés (schedule vagy history): A különböző tranzakciókban lévő műveletek sorrendje, amikor a tranzakciókat összefésülve egy szálon hajtjuk végre.

Szeriális ütemezés: Egy S ütemezést szeriálisnak nevezünk, ha minden ütemezésbeli T tranzakcióra fennáll, hogy az összes T-beli művelet közvetlenül egymás után hajtodik végre az ütemezésben.

Szerializálható ütemezés: Egy S ütemezés szerializálható, ha ekvivalens ugyanazon tranzakciók egy szeriális ütemezésével.

A konkurencia kontroll célja:

- Az elkülönítés kikényszerítése (pl. teljes kizárással) a konfliktusos tranzakciók között.
- Az adatbázis konzisztenciájának megőrzése a tranzakciók konzisztenciamegőrző végrehajtása révén.
- Az olvasás-írás és írás-írás konfliktusok feloldása.

Literál: A literálnak van aktuális értéke viszont nincs (objektum) azonosítója. A literálok háromféle típusa:

- atomi: előredefiniált, alapvető adattípus érték (pl. short, float, boolean, char)
- strukturált: olyan érték, amelyet típus konstruktorral állíthatunk elő (pl. date, struct)
- kollekció: értékek vagy objektumok egy kollekciója (pl. array)

Atomi objektum: Egy a felhasználó által a class kulcsszóval definiált objektum.

Extent: Egy ODMG objektumnak lehet extent-je, amelyet az osztály deklarációsakor definiálunk. Minden extent-et egy névvel adunk meg, és tartalmazza az abba az osztályba tartozó összes perzisztens objektumot.

Pl.: a Dolgozo osztálynak az extentje a mind_dolgozo. Ez hasonló ahhoz, mintha létrehoznánk egy Set<Dolgozo> objektumot és perzisztensé tennénk.

Objektum-orientált (OO) adatbázis rendszerek:

- Object Databases: Objektum-orientált adatbázisok.
- Object Database Management Systems (ODBMS): Objektum-orientált adatbázis-kezelő.
- Fő jellemző: a tervező specifikálni tudja mind a sokszorosán összetett objektumok szerkezetét, mind az ezeken az objektumokon alkalmazható műveleteket.

Objektum: Két komponense van: állapot (érték) és tulajdonság (művelet) (korai definíció). Hasonló a program változójához kivéve, hogy összetett adatstruktúra is lehet és a programozó speciális műveleteket definiálhat rajta. Az objektumok az OO nyelvekben tranziensek (átmenetiek), a futás befejeztével törlődnek. Az objektumok az ODBMS-ekben perzisztensek (állandóak), eltárolódnak, később kinyerhetők és megoszthatóak más programokkal. Egy ODBMS általában több OO programnyelvhez kapcsolódik állandó és osztott objektumokat szolgáltatva számukra.

Objektum azonosító: Az adatbázisban tárolt minden különböző objektum kap objektum azonosítót (OID). Feladata: az adatbázis objektumainak egyértelmű azonosítása, segítségével objektumok közötti hivatkozásokat tudunk létrehozni és kezelni. Egy program alkalmas típusú változójához is hozzárendelhetjük ha szükséges. Tulajdonságok: megváltozhatatlanság, egyértelműség, láthatatlanság (külső felhasználó szempontjából), ábrázolás hosszú egész.

Objektumok összehasonlítása:

- Erős egyenlőség: két objektum azonos állapotban van ha a gráfjaik minden lehetséges szempontból (pl. az OID-k minden szinten) megegyeznek.
- Gyenge egyenlőség: Két objektum egyenlő állapotban van ha a gráfjaik szerkezete megegyezik és a gráfban lévő atomi értékek is megegyeznek (de pl. a gráf közbenső szintjein az OID-k már nem feltétlen)

Konstruktorok:

- (i, atom, érték): az i azonosítójú objektum állapota a v atomi értéket veszi fel.
- (i, halmaz, { $i_1, i_2, \dots, i_n$ }): az i azonosítójú objektum egy halmaz, amely az $i_1, i_2, \dots, i_n$ (általában azonos típusú) objektumokból áll.
- (i, rekord, $\langle a_1:i_1, a_2:i_2, \dots, a_n:i_n \rangle$): az i azonosítójú objektum egy n hosszú rekord, melyben a_j a j-edik attribútum neve, i_j pedig annak az objektumnak az azonosítója melyet értéknek felvesz.
- (i, lista, [$i_1, i_2, \dots, i_n$]): az i azonosítójú objektum egy rendezett lista (azonos típusú) a felsorolásbeli azonosítójú objektumokból.
- (i, tömb, [$i_1, i_2, \dots, i_n$]): az i azonosítójú objektum egy egy-dimenziós tömb a listabeli azonosítójú objektumokból.

Konstruktorok összehasonlítása: A lista abban különbözik a halmaztól, hogy rendezett. A tömb abban különbözik a listától, hogy a dimenziója véges (ismert). A zsák abban különbözik a halmaztól, hogy ugyanaz az objektum többször is szerepelhet.

Objektum-orientált adatbázis-kezelő (ODBM) szabvány részei:

- Objektum modell
- Objektum leíró nyelv (ODL – Object definition language)
- Objektum lekérdezési nyelv (OQL – Object query language)
- Kötések (bindings) az OO nyelvekhez (C++, SmallTalk, Java)

Object Factory: Az egyedi objektumok előállítását az object factory (objektumgyár) segítségével, annak műveletei révén történik.

ODMG interfész: Az interfész egy objektum típus absztrakt viselkedésének specifikációja. Az interfész állapotának tulajdonságai, így attribútumai és kapcsolatai, nem öröklhetőek. Objektumok nem példányosíthatóak interfészen keresztül.

Object Definition Language (ODL): Az ODL támogatja az ODMG szemantikus konstrukcióit. Független bármely programozási nyelvtől. Arra használjuk, hogy létrehozzunk objektum specifikációkat, ún. osztályokat és interfészeket. Nem használható adatbázis manipulációk végrehajtására.

Object Query language (OQL): Az OQL a DMG (Data Management Group) lekérdező nyelve. Az OQL szorosan együttműködik olyan programozási nyelvekkel, mint a C++. A beágyazott OQL utasítások olyan objektumokkal térnek vissza, amelyek kompatibilisek a hívó nyelvrendszer típusaival. Az OQL szintaxisa hasonló az SQL-éhez kiegészítve az objektumok további jellemzőivel.