




JDBC 2



Témák

- Adatbázis tervezés E/R diagram
 - Java DB
 - 0. feladat
- 



Adatbázis tervezés

- ▶ Egy jó adatbázis...
- ▶ ...nem tartalmaz felesleges redundanciát, ugyanaz az adat nem szerepel több helyen.
- ▶ ...biztosítja az adatok épségét és pontosságát.
- ▶ Tervezéskor a feladat az alkalmazás által használandó adatok táblákba rendezése és kapcsolataik definiálása.



ER Diagram

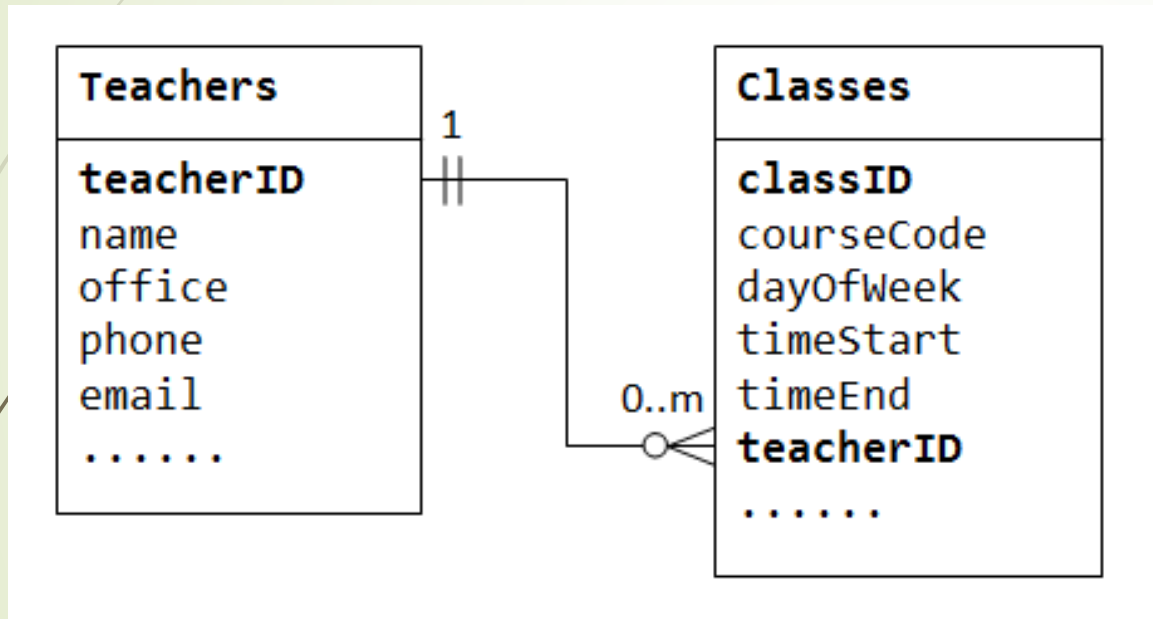
- Az adatbázis logikai modellje egyedkapcsolat diagrammal írható le.
- A diagram elemei:
 - Entitások
 - Attribútumok
 - Kulcsok
 - Kapcsolatok
 - Egy-egy, egy-sok, sok-sok



Példa

- ▶ Hogyan reprezentálható:
Egy tanár taníthat nulla vagy több osztályt, de egy osztályt pontosan egy tanár tanít.
- ▶ **1. lehetőség:** Induljunk ki a tanár táblából, ahol tároljuk az adatain túl a tanított osztályokat: osztály1, osztály2,...
Probléma: szükséges oszlopok száma?
- ▶ **2. lehetőség:** Induljunk ki az osztály táblából, adjuk hozzá a tanár adatait tartalmazó oszlopokat
Probléma: a tanár adatai többször szerepelnek

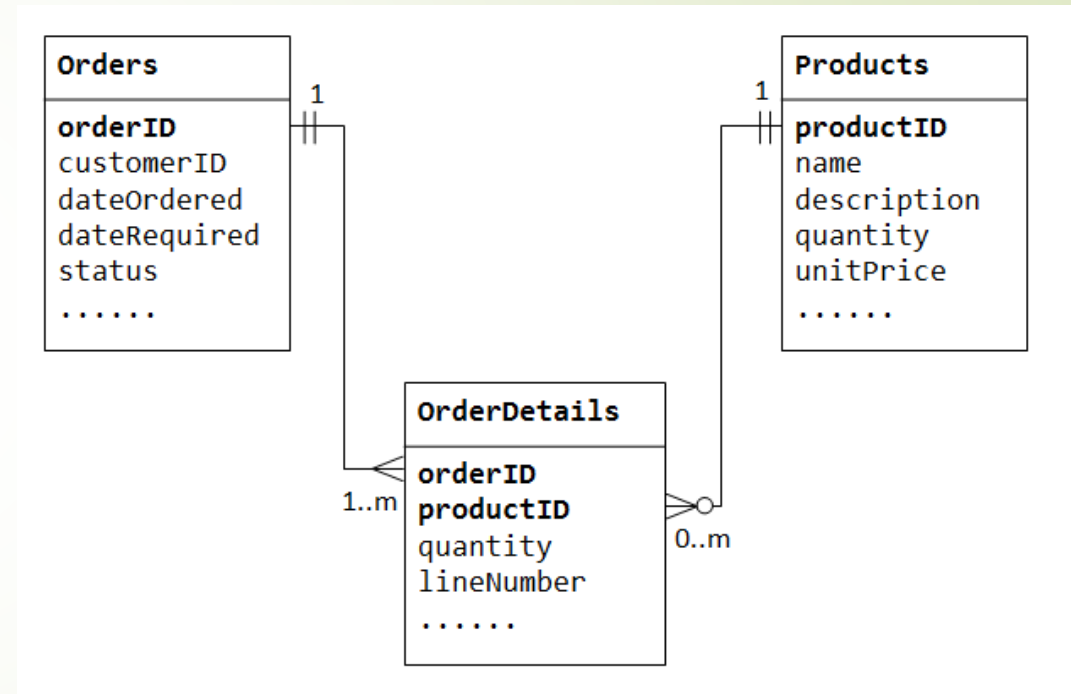
Egy-sok kapcsolat



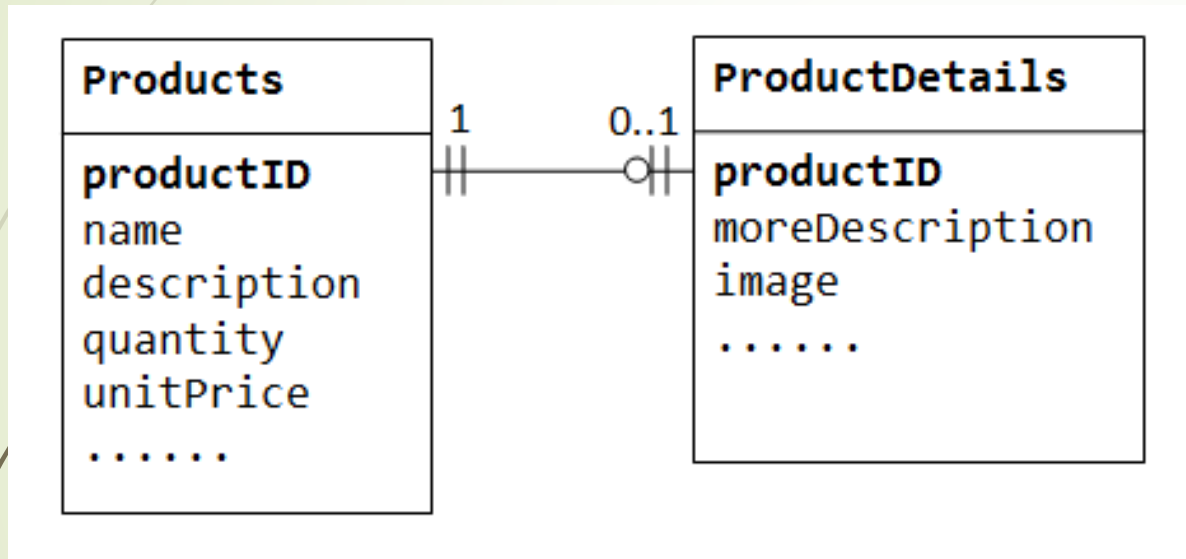
- Egy tanár taníthat nulla vagy több osztályt, de osztályt pontosan egy tanár tanít
- Egy vásárló több rendelést is leadhat, de egy rendelés pontosan egy vevőhöz tartozik.

Sok-Sok kapcsolat

- Egy megrendeléshez egy vagy több termék tartozik
- egy termék több megrendelésben is szerepelhet.



Egy-egy kapcsolat



- Egy termék adatbázisban a termékeknek lehetnek kiegészítő, opcionális adatai.
- A termékekkel egy táblában tárolva számos üres mezőt eredményeznének



ER diagram lépései

- Entitások azonosítása (Az alkalmazás működésében résztvevő „dolgozók”: megrendelés, termék, stb.)
- Kapcsolatok definiálása (Entitások közötti logikai kapcsolat)
- Számosság (Kapcsolatokban hány entitás vehet részt? Egy-egy, sok-egy.. Kötelező?)
- Elsődleges kulcsok megadása (Az adott jellemzők, amelyek egyértelműen azonosítanak egy entitást)
- Sok-Sok kapcsolatok átírása sok-egy kapcsolattá (kapcsolótáblák)
- Attribútumok definiálása és entitáshoz rendelése (Entitások egyéb tulajdonságai, melyik entitáshoz tartozzon?)



JavaDB

- ▶ Apache Derby open source adatbázis amely a JDK részeként elérhető.
- ▶ Konfiguráció és használat:
 - ▶ <https://netbeans.org/kb/docs/ide/java-db.html>
 - ▶ <https://db.apache.org/derby/>



Adattípusok

- ▶ Elérhető – a byte kivételével – az összes java primitívnek megfelelő adattípusa
- ▶ Valamint: Date, Time, Timestamp, String – nek megfelelő típus.

Tábla létrehozása

```
CREATE TABLE table_name (  
    id INT not null primary key  
        GENERATED ALWAYS AS IDENTITY (START WITH 1, INCREMENT BY 1),  
    keszlet INT NOT NULL CONSTRAINT Q_CK CHECK (QUANTITY>=0),  
    elerhető INT NOT NULL CONSTRAINT AV_CK CHECK (AVAILABLE>=0),  
    valami BOOLEAN NOT NULL default false,  
    CONSTRAINT AV_CK2 CHECK (QUANTITY >= AVAILABLE),  
    idegen_kulcs_ID INT NOT NULL references MASIK_TABLE(ID),  
);
```

JDBC generált értékek

- Insert után az automatikusan generált értékek elérésének engedélyezése:

```
PreparedStatement stmt =
    conn.prepareStatement(getInsertSql(), Statement.RETURN_GENERATED_KEYS);
try (ResultSet generatedKeys = stmt.getGeneratedKeys()) {
    if (generatedKeys.next()) {
        t.setId(generatedKeys.getInt(1));
    } else {
        throw new SQLException("Creating entity failed, no ID obtained.");
    }
}
```

Szekvenciák

- Szekvencia egyedi értékek generálására használható, új érték generálása esetén a szekvencia értéke növekszik és független a tranzakciók sikerességétől.

- `CREATE SEQUENCE update_no_seq AS BIGINT START WITH 1;`

- Használata

```
UPDATE orders
```

```
    SET update_no = NEXT VALUE FOR update_no_seq
```

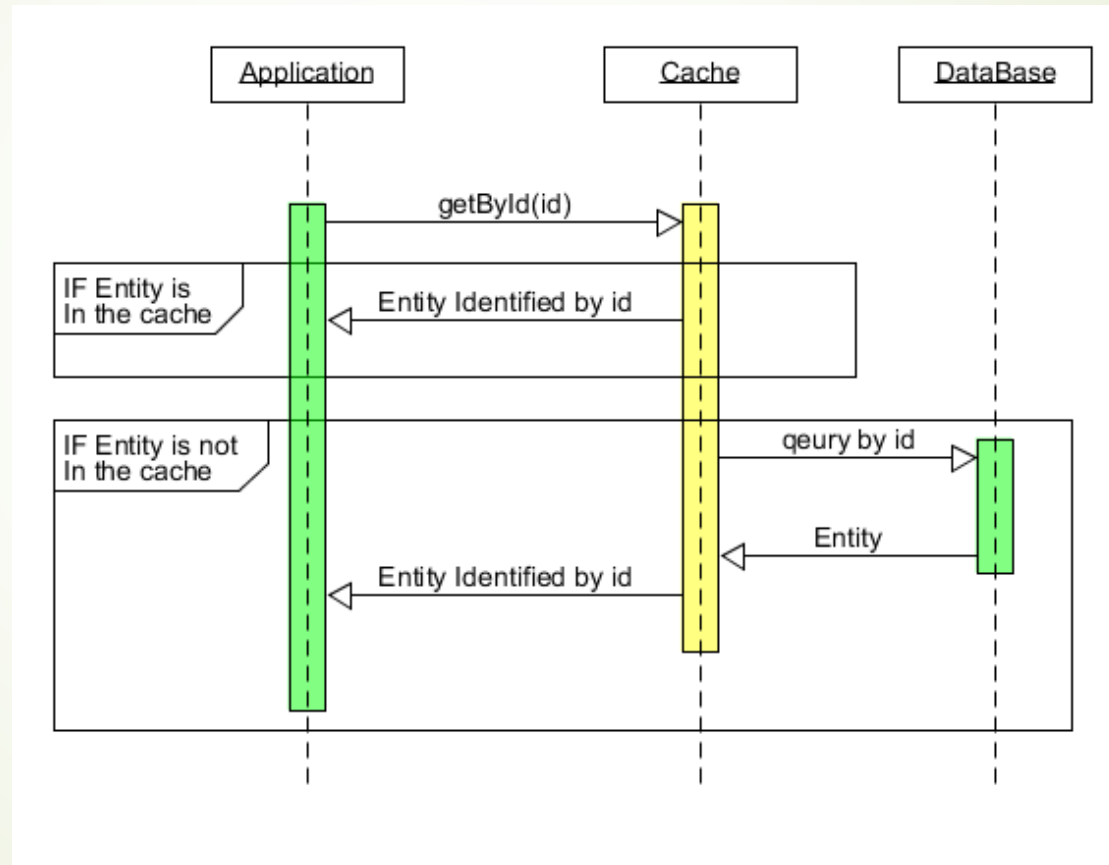
```
    WHERE ..... ;
```



Cache

- ▶ Feladata: egy lokális tároló a logikai réteg és az alkalmazás között. Tartalmazza az adatbázisból betöltött entitásokat.
- ▶ A lekérdezéseket akkor küldi az adatbázisnak, ha nem találhatóak meg lokálisan.
- ▶ Kliens indításkor preload
- ▶ Az új vagy módosult entitásokat periódikusan lekérdezi.

Entitás lekérdezése a lokális tárolóból

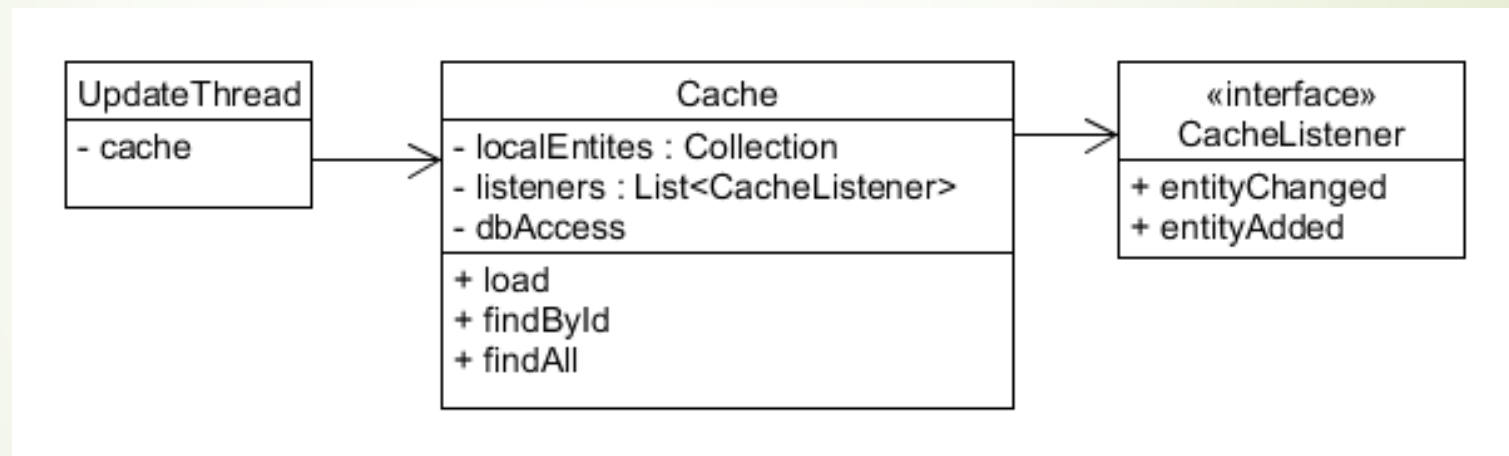




Új, módosult entitások

- ▶ Egy háttérzál valamilyen időzítés, timer segítségével lekérdezi az új elemeket.
- ▶ Új elemeket megadhatjuk az id-juk segítségével. (`..where id > last_id`)
- ▶ Módosult elemek megadhatóak egy módosítás-azonosító, vagy dátum segítségével. (`...where update_key > last_update`)

Minta felépítés





Problémák

- Szálkezelés: lekérdezések és UI frissítések az update szálról
- Konkurens adatbázis használat: pl.: másik alkalmazási kikölcsonzi előlünk a dolgokat.
- A szerkesztett elemet módosítja másik alkalmazás



Vége

