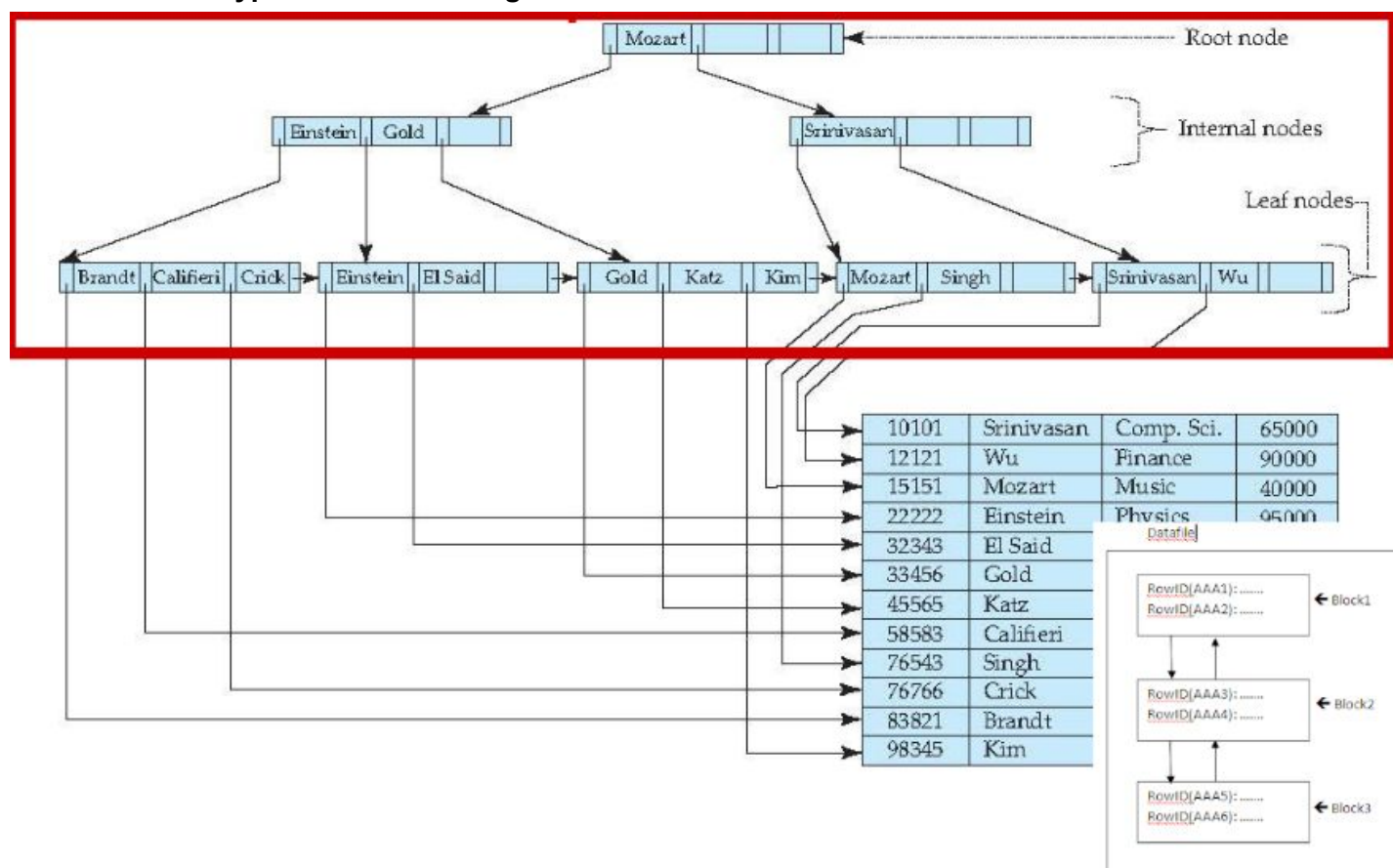


# Database Advanced - 1. Exam 2019 Solved

A

- Draw a figure showing the structure of a B+ index and the corresponding table!
- Mark the important components in the figure!
- Mark the typical sizes in the figure!



Csak a teljesség kedvéért:

A fa egy gyökérellemmel (root) rendelkezik, ami alatt több szinten köztes csomópontok találhatóak, legalul pedig levelek (leaf) helyezkednek el. Az ábrán egyes csomópontok 1-1 blokkot jelölnek. A gyökér és a köztes csomópontok azonos felépítésűek, kulcs-mutató párokból állnak. A levelekben kulcs, rowID párok találhatóak, alapesetben a tábla minden sorára egy ilyen pointer mutat.

Typical size of a node ~ block on disk (4 / 8 / 16 / 32 kb)

**Which size of a B+ tree determines the cost of a query (let us consider a simple case, when the query is very selective, returns only a few records)?**

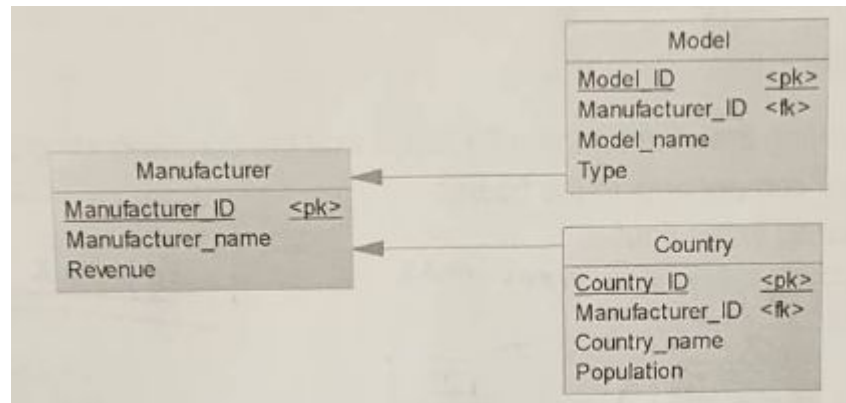
The height of the B+ tree determines the cost of a query. (Ezt nem tudom)

**What can we say about index usage in case of a query with bad selectivity (high ratio of rows returned)? Please explain your answer!**

Index is useful for a query condition with at most ~5% selectivity. In case of bad selectivity index usage doesn't make the query cost less. However if the index is composed for the selected query it can return the records much faster.

## B

The following relational schema diagram is given, describing bike manufacturers, their bike models and the countries in which they have representatives.



Write an SQL statement, that provides some basic statistics for the manufacturers, with the following columns:

| Manufacturer_name | Sum of population<br>(sum of population of countries in which the manufacturer has a representative) | Number of models<br>(number of models the manufacturer produces) |
|-------------------|------------------------------------------------------------------------------------------------------|------------------------------------------------------------------|
|-------------------|------------------------------------------------------------------------------------------------------|------------------------------------------------------------------|

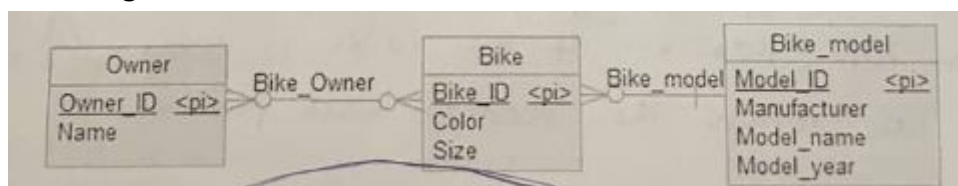
Soha nem értettem sajna rendesen ezeket a joinokat, de ez csinál valamit az biztos, csak max köze nincs a feladathoz.

```

SELECT Manufacturer.Manufacturer_name, SUM(Country.Population),
COUNT(Model.Model_ID)
FROM ((Manufacturer
INNER JOIN Model ON Manufacturer.Manufacturer_ID = Model.Manufacturer_ID)
INNER JOIN Country ON Manufacturer.Manufacturer_ID = Country.Manufacturer_ID)
    
```

## C

Create a relational diagram for the following ER diagram. Mark primary and foreign keys clearly! Use the relational schema diagram notation of task B!



Many to many relationshippeknél csinálsz egy tök új táblát, amiben a két oldal primary keyei vannak ( Related\_to(Card.gamepcs\_id, Event.event\_id))

One to Many-nél vagy Many to One-nál, a Many oldalon tárolod a One oldal PK-ját.

Amikor valaki más PK-ját csak így tárolgatod na az egy FK. Ennyi igazából. Ennek mintáján megrajzolod az előzőt. Ha jól értem, amiből FK-t tárolsz, arra kimegy belőled egy nyíl. Igazából ez ordas nagy tipp, de 1-2 pontot biztos megér.

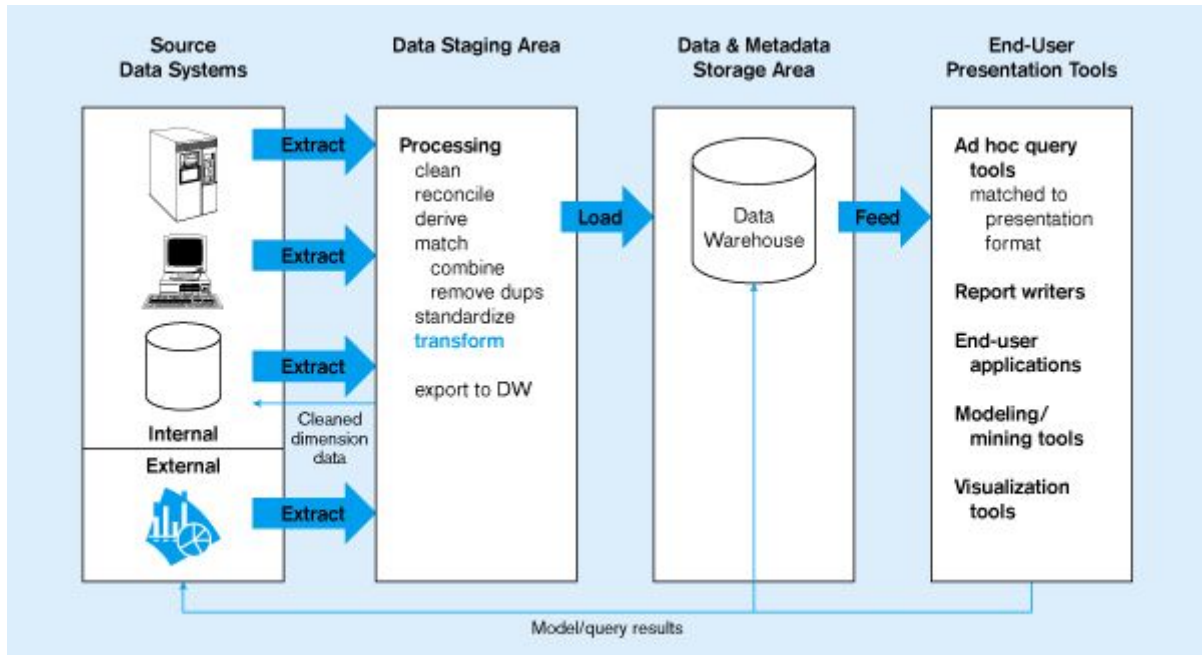
# D

## Explain major concepts of data warehousing!

### Motivation, definition:

A copy of transaction data, specifically arranged, grouped from operational data sources, to support business decisions, reporting etc.

### Architecture, major elements:

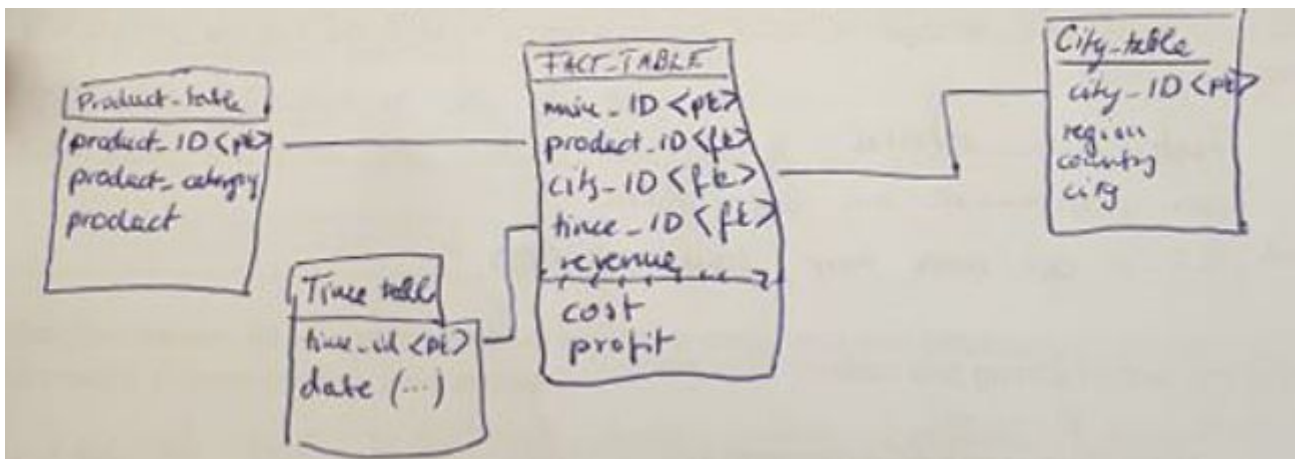


### Challenges (some):

wat?

Draw a star schema for the following case! (Use the relational schema diagram notation in task ...!)

- Measures: revenue, cost, profit
- Dimension 1: region, country, city
- Dimension 2: product category, product
- Dimension 3: date (year, quarter, month, week, day)



Possibilities to improve performance / optimizer a data warehouse (in a wide sense)?

- Star schema (Bitmap index)
- Star transformation
- Materialized view, query rewrite

# E

**What advantages / disadvantages has Hibernate over JDBC? (Mark advantages with + sign and disadvantages with - sign!)**

- (+) Egyszerűbb, gyorsabb alkalmazás fejlesztés
- (+) Könnyebben karbantartható kód
- (+) Cachelés
- (+) Adatbázistól független fejlesztés
- (-) Futásnál lassabb, nyilván kevésbé optimalizált, mint a JDBC

**Compare object-relational mappers and object-relational databases! What are the typical scenarios for their usage?**

Az objektumrelációs adatbázis-kezelők legfőbb elemei:

1. Extensible data types
2. Methods / operations
3. Extensible indexing

Jellemző használati területei:

- Geographical data / Spatial database(Maps) (a legfontosabb) - használt program: PostgreSQL
- Multimedia/complex data types eg. XML - használt program: Oracle

Objektum relációs leképezés: Az ORM egy programozási technika, ami objektum orientált rendszert hivatott leképezni egy nem kompatibilis rendszerbe (pl. relációs adatbázis). A cél, hogy az objektumok tulajdonságait és kapcsolatait megőrizzük.

**What advantages / disadvantages has a relational database over the Python Pandas package? (Mark advantages with + sign and disadvantages with - sign!)**

- (+) Better performance for queries
- (-) Does not have a toolkit for displaying the data by default
- (-) Pandas can be used for machine learning, deep learning etc.
- (-) Pandas esetén könnyebb fejlesztés és debuggolás
- (-) Pandas egy egyszerűbb dolog főleg a python miatt
- (+) Viszont a pandas sokkal több erőforrást használ (pl. memória)

**Compare NoSQL databases (key-value/document stores, MongoDB) to relational ("SQL") databases! (Mark the advantages of NoSQL systems with + sign and disadvantages with - sign!)**

- (+) Good for constantly changing data
- (+) Store very large amount of data
- (+) Faster development
- (-) Less optimizer for complex queries, other database tasks
- (-) Relational databases are better when there are strong relations between the stored datas.

**What is the cost of joining two tables?**

- **Table R**
  - **T(R): number of tuples (records) in R**
  - **P(R): number of pages in R**
- **Table S**
  - **T(S): number of tuples in S**
  - **P(S): number of pages in S**
- **Ignore the cost of outputting the result!**

**Cost of nested loop join, outer table R:**

$$P(R) + T(R) \cdot P(S)$$

**Cost of block nested loop join, outer table R (B pages are available in the memory):**

$$P(R) + \frac{P(R)}{B-1} P(S)$$

**Cost of block nested loop join outer table S (B pages are available in the memory):**

Mint az előző csak S meg R megcserélve