1 Database management systems

Databases are developed in a way to grant data independence, integrity and security, efficient and concurrent access and reduced application development time.

1.1 Data models

A data model are the concepts used to describe data. A schema is the description of a particular collection of data.
The most used model today is the relational model. It bases its self on the concept of relations, and every relation has a schema

1.1.1 Abstraction

Data is defined in different schemas. The physical schema represents how the data is stored on the disk, the conceptual schema defines the logical structure, and a view is how users see the data.

Schemas are defined using DDL (Data Definition Language) and the data is modified using DML (Data Manipulation Language)

1.2 ACID

ACID is the list of properties that a database must ensure. To explaine them we use the example of a bank transaction

1.2.1 Atomicity

Atomicity is a property caracterized by the all-or-nothing policy. It is applied through a log who keeps

1.2.2 Consistency

The databse must be consistent before and after the transaction

1.2.3 Isolation

multiple transactions occur independently without interference

1.2.4 Durability

THe changes of a successful transaction occurs even if the system failure occurs

2 E/R Model

2.1 Entity

an entity is a real world object which is described using a set of attributes. A collection of similar entities is a entity set

weak entities

This are entities that depend to another entity. The weak entity doesn't make sense to exist by itself. This entities are rappresented through a thick line around the entity and they use a thick arrow towards the entity they depend

2.2 Relations

It is a connetion between two or more entity sets.
The relation can use different multiplicity, which are many to many, many to one and one to one. We use the wisconsin representation system, which is structured like this:
we use an arrow to point to "one", a think arrow to represent "exactly one", a line to represent "mmany to many" and a thick line for "at least one"

2.2.1 Aggregation

is used when we have to model a relationship involving entity sets and a relationship set

2.3 Subclasses

it's a special case of subentity, it's represented with a triangle

E/R Subclasses

Subclasses form a tree of one-one inheritance that use isa relationships.

2.4 Keys

This are unique identifiers of an entity and are represented through an underlined attribute. They can be super keys (keys in general), candidate key (minimal super key) and primary key (the chosen key)

2.5 Schemas

3 Relational model

it's the most used model thanks to the characteristic to be easier and faster than other models.

Definitions

a relational database is a set of relations, which are made of instances, or tables, and schemas, which specifies name of the relation, and name plus type of each column. In the tables, rows equals to the cardinality and are called tuples,the fields are called degrees

3.1 Integrity contraints

A condition must be true for any instance of the database.

Keys can be contraints because no distinct tuples can have the same values

4 Relational algebra

The manipulation and retrieval of data from a database is managed through query language. A mathematical query language i the relation algebra, an operational query language.

The basic operations are:

- Selection (σ): selects a subset of rows from relation
- Projection (π): selects the columns from relation
- Cross-product ($\times$): allows to combine two relations
- Set difference (-)
- Union ($\cup$)

there are also other sub-operators:

- Renomination (ρ, $\rightarrow$)
- Join ($\bowtie$) or theta join: the union on determined conditions
- Equijoin: a join on only equalities
- Natural join: is a equijoin on all common fields

5 SQL

A basic SQL query is composed by relation-list, target-list and qualification and it follows this strategy:

- Compute the cross-product of relation-list (FROM)
- discard resulting tuples if they fail qualifications (SELECT)
- delete attributes that are not in target-list (WHERE)

SQL is also capable of nesting queries. This is done by adding the operator IN in the WHERE section, and the nested query inside. It can be also used the operator EXISTS to let the inner query use parameters from the outer query. parentesis

5.0.1 Operators LIKE and AS

LIKE is used in string matching, it uses % to represent multiple characters and _ just for one. To generate new name fields is also used the operator AS

5.0.2 UNION and INTERSECT

UNION can be used to unite two compatible sets of tuples, meanwhile INTERSECT computes the intersection between the tuples

5.0.3 Agregate Operators

SQL offers various extentions to relational algebra, such as COUNT, SUM, AVG, MAX and MIN

5.1 Deductive Databases

TO apply recursevly

6 Indexing

On datastorage, Disk can retrieve random pages at fixed cost, but must be organazied. This is called file organization, or the mothod of arranging a file of records. Indexes are the data structures that allow to find records ids.Indexes are organized through trees or hashing.
Alternatively, other file organizations can be heap files or sorted files.

An index contains a collection of data entries $x*$ with a key value of x. Any field of a relation can be the search key for an index.

6.0.1 B+ tree indexes

Data entries are contained in the leaves of a tree and are chained to one another. Non-leaf pages have index entries used to direct searches

6.0.2 Hash-based indexes

Better suited for equality selection, this indexes are a collection of buckets (primary pages with zero or more overflow pages) which contain data entries. It's also implemented an hashing function that retrieves the buckets where a record belongs.

6.1 Alternatives for data entry x in index

we can store the data record with key value x, rids of data record or list of rids of data records.

In the first alternative, the index structure is a file organization of data records. At most one index on a collection of data records can use this alternative, otherwise the data is duplicated. If data records are very large, the number of pages is high, this implies also lage size of the auxiliary information

With the other alternatives data entries are much smaller than data records, so more efficient for large data records. The third alternative is even more compact

6.2 Clustered and unclustered indexes

An index can be categorized in primary vs secondary, and clustered vs unclustered. These second classification differs for a sorted or unsorted heap file with all the data records

Page = P tuple record = Tr space of R = Pr = #P cardinality of R = —R— cardinality of attribute (distinct) = —R.a—