DATABASE

A database is a repository of data, designed to support efficient data storage, retrieval and maintenance.

If data is stored in a tabular form, then it is called a relational database

When data is organized in a tree structure form, it is called a hierarchical database

Data stored as graphs representing relationships between objects is referred to as a network database

Database management system [DBMS] is a set of software tools that control access, organize, store, manage, retrieve and maintain data in a database

An information model is an abstract, formal representation of entities that includes their properties, relationships and the operations that can be performed on them.

Another important characteristic of an Information Model is that it defines relationships between managed objects.

Data Models, on the other hand, are defined at a more concrete level and include many details

A data model is the blueprint of any database system

A data architect is responsible for designing an architecture that supports the organization's existing and future needs for data management

A database administrator (DBA) is responsible for the maintenance, performance, integrity and security of a database

A database application developer is a person in charge of developing applications that access databases

Domain is a set of atomic values that are all of the same type

Value is the smallest unit of data in the relational model

Tuple is an ordered set of values that describe data characteristics at one moment in time.

A relation degree is equivalent with the number of attributes of that relation

Relation cardinality is equivalent with the number of tuples of that relation

The body consists of a time-varying set of tuples.(i.e the Relation cardinality varies with time)

The state in which a relation exists at one moment in time is called a relation instance. Therefore, during a relation's lifetime there may be many

relation instances.

The four properties for all Relations are as follows:

?? There are no duplicate tuples in a relation

?? Tuples are unordered (top to bottom)

?? Attributes are unordered (left to right)

?? All attribute values are atomic

Informal terms used for relations are table or data file.

A Database Schema is a formal description of all the database relations and all the relationships existing between them.

A Candidate Key is a unique identifier for the tuples of a relation. By definition, every relation has at least one candidate key (the first property of a relation).

A key is said to be Candidate key if and only if it satisfies the Uniqueness and Minimality Properties.

A candidate key is sometimes called a unique key.

If a relation has more than one candidate key, the one that is chosen to represent the relation is called the primary key, and the remaining candidate keys are

called alternate keys.

A Primary Key is a unique identifier of the relation tuples.

It is a candidate key that is chosen to represent the relation in the database and to provide a way to uniquely identify each tuple of the relation.

DDL - Data Definition Language

Surrogate Key - This is a key with no meaning to the real-word data. eg. ID

A Foreign Key is an attribute (or attribute combination) in one relation R2 whose values are

required to match those of the primary key of some relation R1 (R1 and R2 not necessarily distinct).

Note that a foreign key and the corresponding primary key should be defined on the same underlying domain.

In a relational data model, data integrity can be achieved using integrity rules or constraints

If a user attempts to execute an operation that would violate the constraint then the system must then either reject the operation or in more complicated

situations, perform some compensating action on some other part of the database.

The relational data model constraints are:

1. Entity integrity constraint - It says that no attribute participating in the primary key of a

relation is allowed to accept null values.

A Null represents property inapplicable or information unknown.

Primary keys perform the unique identification function in the relational model

2. Referential integrity constraint - It says that if a relation R2 includes a foreign key FK matching the primary key PK of other relation R1, then every

value of FK in R2 must either be equal to the value of PK in some tuple of R1 or be wholly null (each attribute value participating in that FK value must be null).

R1 and R2 are not necessarily distinct

What should happen on an attempt to delete the primary key value tuple of a foreign key reference?

In general, there are three possibilities:

a. CASCADE – the delete operation “cascades” to delete those matching tuples also (the tuples from the foreign key relation). In our case, if the car

is deleted the owner is deleted, too.

b. RESTRICT - the delete operation is “restricted” to the case where there are no such matching tuples (it is rejected otherwise). In our case, the car can be

deleted only if it is not owned by a person.

c. NULLIFIES – the foreign key is set to null in all such matching cases and the tuple containing the primary key value is then deleted

(of course, this case could not apply if the foreign key cannot accept null values).

N.B: The above possibilities are also apply to an attempt to update the primary key value of a foreign

key reference.

3. Semantic Integrity Constraint - This refers to the correctness of the meaning of the data. It can be regarded as a predicate that all correct

states of relations instances from the database are required to satisfy.

Domain constraint implies that a particular attribute of a relation is defined on a particular domain.

A domain constraint simply states that values of the attribute in question are required to belong to the set on values constituting the underlying domain.

A null constraint specifies that attribute values cannot be null. It is usually specified with the NOT NULL construct.

A unique constraint specifies that attribute values must be different. It is not possible to have two tuples in a relation with the same values for that attribute.

Relational algebra is a set of operators to manipulate relations.

A NULL is part of the UNIQUE data values domain

A table whose column values depend on the values of other tables is called dependant, or child table

A table that is being referenced is called the base or parent table

A VIEW is a virtual table consisting of different columns from one or more tables or other views. It is virtual because it does not contain any data,

but a definition of a table based on the result of a SELECT statement.

Unlike a table, a view is stored in the database as a Query Object. Therefore, a view is an object that obtains its data from one or more tables

Views allow you to hide data or limit access to a select number of columns; therefore, they can also be used for security purposes.

To create a view:

CREATE VIEW MYVIEW AS

SELECT LASTNAME, HIREDATE FROM EMPLOYEE

Selecting data in SQL is an operation that allows you to read (retrieve) rows and columns from a relational table

A model is an abstraction or representation of the real world that reveals all the features of interest to the users of the information in the model

Data modeling defines all the required information from the real world.

A database model is used to represent data about data, also known as metadata. A database model is used for a database schema description.

Conceptual data model is a mental image of a familiar physical object and are not specific to a database.

An entity set is a set of entities of the same type that share the same properties

An entity is an instance of an entity set.

Candidate key – A candidate key is an attribute or set of attributes that uniquely identifies a record in a relation.

Primary key – A primary key is one of the candidate keys from a relation. Every relation must have a primary key.

Cardinalities – restrictions on which attribute values and which relationships are allowed

They are based on the number of possible relationship sets for every entity set. Examples: one to one, one to many, many to many

TYPES OF ATTRIBUTE

Simple (atomic) attribute – This type of attribute has a single component. For example, the Gender attribute has a single component with two values.

?? Composite attribute – A composite attribute consists of many components. For example, the Name attribute has the components last name and first name.

?? Single valued attribute – This type of attribute has one value for one entity. For example, the Title attribute has a single value for each teacher.

?? Multi-valued attribute – A multi-valued attribute has many values for one entity. For example, a person has many phone numbers. Each attribute can have only one value for each instance of the entity set. When you encounter a multi-valued attribute you need to transfer it to another entity set.

?? Derived attribute – A derived attribute has its value computed from another attribute or attributes. For example, the sum of all students in a faculty. A derived attribute is not a part of a table from a database, but is shown for clarity or included for design purposes even though it adds no semantic information; it also provides clues for application programmers.

?? Unstable attributes - This type of attribute have values that always change. For example, the study year of a student.

?? Stable attributes - Stable attributes will change on the odd occasion, if ever.

?? Mandatory attributes - Mandatory attributes must have a value. For example, in most businesses that track personal information, Name is required.

Types of Constraint at Conceptual Model

1. Cardinalities
2. Optionalities
3. Supersets and Subsets
4. Unary relationship set
5. History

An entity set is weak if its existence lays on the existence of another entity set.

An entity set is strong if its existence is an independent one

Developing the conceptual model

Step 1 - Identify entities

To identify an entity set, you need to review the requirements specification carefully, and highlight all nouns you encounter.

Step 2 - Remove duplicate entities

Don't include the system as an entity set.

Step 3 - List the attributes, type and domain of each entity set

- 1. Make a decision about composite attributes: leave it or split it

- 2. When you encounter a multi-valued attribute, you need to transfer it to another entity set

Step 4 - Choose a unique identifier

Step 5 - Define the relationship sets

- In this step you need to examine the relationships between the different entity sets

- Describe the cardinality and optionality of the relationship sets you find, and remove redundant relationship sets.

Step 6 - Define business rules

- Business rules have to be implemented by an application program

An identifying relationship set is selected to specify that the relationship set is one in which one of the child entities is also a dependent entity.

Non-Identifying relationship set is selected to specify that the relationship set is one in which both entities are independent.

The CHECK clause in the SQL standard allows domains to be restricted

Data in a database at a particular point in time is called an Extension of the database. Extension refers to the current set of tuples in a relation

Intension or schema is the logical model of a database and is represented by entity sets.

Data redundancy implies finding the same data in more than one location within database tables

Functional Dependency (FD) is a type of integrity constraint that extends the idea of a super key.

It defines a dependency between subsets of attributes of a given relation.

Normalization is a procedure in relational database design that aims at converting relational schemas into a more desirable form.

The goal is to remove redundancy in relations and the problems that follow from it, namely insertion, deletion and update anomalies.

A relation is considered to be in FIRST NORMAL FORM if all of its attributes have domains that are indivisible or atomic.

The idea of atomic values for attribute ensures that there are no ‘repeating groups’ .

A relation is in SECOND NORMAL FORM when it is in 1NF and there is no such non-key attribute that depends on part of the candidate key, but on the entire candidate key.

It follows from the above definition that a relation that has a single attribute as its candidate key is always in 2NF

A relation is in 3RD NORMAL FORM if it is in 2NF and there is no such non-key attribute that depends transitively on the candidate key.

That is every attribute depends directly on the primary key and not through a transitive relation where an attribute Z may depend on a

non-key attribute Y and Y in turn depends on the primary key X.

It follows from 3NF relation that the non-key attributes are mutually independent.

Boyce-Codd Normal Form is a stricter version of 3NF that applies to relations where there may be overlapping candidate keys.

A relation is said to be in Boyce-Codd normal form if it is in 3NF and every non-trivial FD given for this relation has a candidate key as its determinant.

That is, for every X ->Y, X is a candidate key.

Decomposition is breaking up a relational schema into smaller relations such that each of the attributes is present in at least one of the new relations

and has a more optimized design.

The underlying goal is to remove redundancy.

A Decomposition is said to be lossless if no information from the original relation is lost after the decomposition.

Entities in a data model generally map to tables when implemented in databases.

Attributes of entities map to columns of the table.

Selecting data in SQL is an operation that allows you to read (retrieve) rows and columns from a relational table

The special character ' * ', represents all the columns from the table

To guarantee the result set is displayed in the same order all the time, either in ascending or descending order of a column or set of columns, use the ORDER BY clause.

Ascending order (ASC) is the default when using ORDER BY

If the WHERE clause is not used, the DELETE statement will delete all rows from the table.

If the WHERE clause is not used, the UPDATE statement will update all rows in the table.

Equi-join - This type of join happens when two tables are joined based on the equality of specified columns; For Example,

Implicit Inner Join

```
SELECT *
FROM student, enrollment
WHERE student.enrollment_no=enrollment.enrollment_no
```

Explicit Inner Join

```
SELECT *
FROM student
INNER JOIN enrollment
ON student.enrollment_no =enrollment.enrollment_no
```

A Natural Join is an improved version of an equi-join where the joining column does not

require specification. For Example,

```
SELECT *
FROM STUDENT
NATURAL JOIN ENROLLMENT
```