

Module 1

Database Management system

A collection of data that contains information relevant to an enterprise is known as database. DBMS is a collection of interrelated data and set of programs to access these data. The primary goal of DBMS is to provide a way to store and retrieve database information that is both convenient and easy efficient.

③ Three tier architecture

- Most widely used
- Follows client - Application - server technique
- It is an improvisation to two tier architecture by adding an intermediate or application layer between the server (database layer) and client (presentation layer) to reduce query processing load of the server. The application layer is responsible for load balancing, query correctness and security.
- Enhance security, scalability and data integrity because of the intermediate application layer.
- Due to increase in no. of layers between client and server, implementation of DBMS structure is complex and difficult to maintain.

Relational Database

Relational databases use a collection of tables to represent both data and the relationships among those data.

Table

Each table has multiple columns with unique column names.

Integrity constraints

- certain set of rules which are used to maintain the quality and consistency of data in the database.
- Every time there is insertion, deletion or updating of data, integrity constraints maintain integrity of data and thus help to prevent accidental damage to the database.
- There are 4 types.

① Domain constraint

contain rules / conditions to restrict the kind of attributes or values a column can hold in the database ~~model~~ table. The datatype of a domain can be string, integer, character, datetime, currency etc.

② Entity Integrity constraint

ensures that primary key does not contain NULL.

③ Referential Integrity constraint

Ensures that there exists a valid relationship between two relations. This valid relationship confirms that a foreign key exists in the table.

④ Key constraint

Keys are the set of entities that uniquely identifies an entity within its entity set. There can be multiple keys. A primary key needs to be unique and not null. (include types of keys).

comparison basis

DDL

DML

Hope design Schema which deals w how data is stored.

Allows to manipulate stored data in the database.

full form

data definition

data manipulation

Categories

NONE

Procedural, declarative

Command

create, drop, truncate,
alter, rename

Select, insert, update,
delete

Auto Commit

Yes. Any changes in
database is permanent

No. Database changes
are not permanent

WHERE clause

Pituation of records
is not possible.

clause is used while
manipulating data.

effect

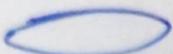
Affects entire database

Affects single or multiple
rows.

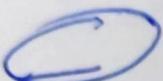
② Entity relationship model
This model uses real world objects and entities and their relationships among them for database design.

Type of Attributes

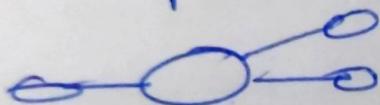
① simple



② key



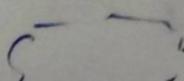
③ composite



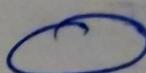
④ Multivalued



⑤ Domined



⑥ single valued



Type of Relationship

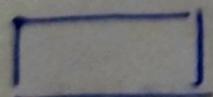


- simple

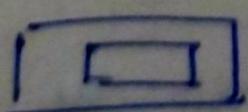


- Identifying relationship. (weak entities)

Type of entities



- entity



- weak entity

ER Model

- Entity relationship model
- Used to create database design using 3 basic concepts - entity sets, relationship sets and attributes

Entity

- An entity is a real world object that is distinguishable from other objects. An entity has a set of properties and the values of some of these properties may be used to uniquely identify ~~an~~ an entity (key attribute).
~~If~~ An entity course has course-id which uniquely identifies it.
- A set of entities of the same type that share the same properties is called an entity set.
~~If~~ set of instructors is an instructor entity set.
- An entity set need not be disjoint.
~~If~~ A person entity may belong to both instructor set and student set.
- An entity set is distributed into a set of attributes each of which contains values describing properties of an entity set. The values of these attributes may be used to identify an entity. The value that uniquely identifies an entity is called the key attribute.

Relationship set

- A relationship is an association among several entities. A relationship set is a set of relationships of the same type. Entity sets participate in relationship sets.
- The function that an entity plays in a relationship is called entity role.

- the same entity set participating in a relationship set more than once, in different roles is called recursive relationship.
eg course A of course entity set is prerequisite for course from course set. Here, relationship set prerequisite is a recursive relationship.
- A relationship set may also have descriptive attributes.
- The number of entity sets that participate in a relationship is the degree of the relationship set.

Attributes

- For each attribute there is a set of permitted values called domain or value set. There are different types of attributes
 - ① Single / multivalued attributes
 - ② Simple / composite attributes
 - ③ Derived attributes
Attribute that can be derived from the values of other related attributes or entities.
- NULL values may indicate
 - ① Not applicable - value does not exist
 - ② Missing - value exists but we don't know
 - ③ Unknown - We don't know whether or not value exists

eg Instructor name = NULL indicates name is missing

eg Apartment no. = NULL in instructor address can mean, no apartment no. (Not applicable), we don't know the no. (missing), the apartment no. may or may not be part of the address (unknown)

Module 2

Relational Query Languages

Relational Model

Primary key: key attribute. There exists only one primary key in a table. A combination of multiple attributes as the primary key is a composite primary key. It is used to uniquely identify a row in a table and cannot be duplicated; cannot be null.

Foreign key: it is the primary key of another table that is used to link two tables together.

Tuple: A tuple refers to a row in a relation.

Attribute: The term attribute refers to the column in a relation.

Relation Instance: it refers to a specific instance of a relation that contains a specific set of rows.

Database instance: it is a snapshot of the data given in the database at any given time.

Database schema: it is the logical design of a database. A database schema does not generally change however instances change with different updates.

Superkey: A set of one or more attributes that uniquely identify a tuple from a relation. All super keys cannot be primary keys.

Candidate keys: superkeys that are minimal.
superkeys for which there are no subsets are called candidate keys.

Primary key: denotes a candidate key that is chosen by the database designer as the principal means to identify tuples in a relationship relation.

Referring relation: The relation that connects a foreign key to another relation is called referring relation of the foreign key.

Referential Integrity constraint: This constraint requires that the values appearing in specified attributes of any tuple in the referring relation also appear in specified attributes of at least one tuple in the influenced relation.

Schema Diagrams: A database schema along with primary key and foreign key dependencies is depicted with schema diagrams.

Query language: It is a language that can be used by the user to request information from the database.

Procedural language: The user instructs the system to perform a sequence of operations on the database to compute the desired result.

Non procedural language: The user describes the desired information without giving a specific procedure to obtain the information.

Aggregate functions

Average = avg

Minimum = min

Maximum = max

Total = sum

Count = count

select avg (column) from table

select min (column) from table

select max (column) from table

select sum (column) from table

select count (column) from table

Viva short answers

- DCL - data control lang
 - grant
 - revoke

controls , rights , permissions
of database system
- TCI - Transaction control
 - Savepoint
 - Rollback
 - commit

maintain consistency &
integrity of database
during transactions
- Modify (id NOT NULL) - adding not null constraint
format
- %TYPE : provides data type of variable or column to
that variable
- %ROWTYPE : provides record type of a row in a table or view
- %ROWCOUNT : stops query processing after specified number
of records are written.
- Raise_application_error - Oracle defined procedure
that helps to raise user defined errors
- char(7) fixed length
Varchar() - does not exist in Oracle
Varchar2() - max length is specified

• Subquery = nested query

query inside another query

• Types of triggers

1) DML Triggers - executed in response to DML statements

2) DDL

3) Instead Of Triggers - replaces the standard action of a DML statement

4) Row-level - triggered once for each row. DML

5) Statement-level - triggered once for each statement. Not dependent on rows
More efficient.

• Mutating triggers?

• Recoverable schedules

Seq of database transactions that can be used to restore database to consistent state

• Cascading schedules

P.T.O

Deadlocks

Order of operations in transactions is determined based on data dependencies rather than the order in which they were submitted.

Deadlocks

Happens in multiuser DBMS when two or more transactions are waiting for each other to be completed.

Timestamp ordering

Ensures transactions are done in correct order based on timestamps assigned to each transaction.

use of host string

A host string is the name assigned for TNS connectivity. These numbers identify machine, port and SID you are trying to connect to.

Two phase locking

Concurrency control technique that guarantees serializability.

Conflict Serializability

Orders any conflicting operations in the same way as some serial execution.

DBMS failures

network failure
system crash
dueller
carelessness
Sabotage
Software error

Date function examples

Now()
cudate()
extract()
date()
date_add()
datediff()

String function examples

ltrim()
rtrim()
lrcat()
lrcopy()

Number function examples

abs()
acos()
asin()
ceil()
Ceiling()
cos()
cot()