

CLUSTER

Software Solutions

“DATABASE”

--By RavikumarHadagali

Table of Contents

1. INTRODUCTION.....	3
2. DBMS(Data Base Management System).....	3
3. RDBMS (Relational Data Base Management System)	3
4. ORDBMS (Object Relational Database Management System)	4
5. CONSTRAINTS.....	7
6. SEQUENCE.....	9
7. NORMALIZATION	9
8. DENORMALIZATION	9
9. SQL FUNCTIONS	10
10. AGGREGATE FUNCTIONS OR GROUPING FUNCTIONS.....	12
11. SQL ALIAS	12
12. STRUCTURE OF SQL QUERY.....	13
13. SQL JOINS.....	15
14. VIEW	17
15. INDEX	18
16. SYNONYM.....	19

INTRODUCTION

- Data is nothing but value.
- Database is a repository and is used to store data (especially related to an organization or enterprise). It is an example for secondary memory.
- Earlier file management system was used for storing and maintaining data. Various problems with file system storage are:-
 1. Doing CRUD operations like insert, read, update and delete was difficult.
 2. Data redundancy or duplication.
- To avoid all the problems with file management system DBMS was introduced.

DBMS(Data Base Management System)

- DBMS is a software management system to maintain database.
- The software required to perform the task of database management is called as DBMS (Data Base Management System).
- Management of data means :
 - 1) Providing mechanisms for data manipulation such as adding, updating, deleting or reading of data.
 - 2) Providing data security against unauthorized access.
- There are many types of DBMS implementation but among which RDBMS and ORDBMS are popular.

RDBMS (Relational Data Base Management System)

- A **relational database management system (RDBMS)** is an implementation of database management system (DBMS) or software that is used to maintain database.
- In RDBMS data will be stored in tabular format.
- In RDBMS data will be stored in many tables and all the tables will be related with each other with the help of Primary Key and Foreign Key.
- RDBMS database support normalization.
- RDBMS database use SQL (Structured Query Language) to communicate with the system.
- If you want to do any CRUD (CREATE, READ, UPDATE and DELETE) operation in RDBMS you have to enter a valid SQL statement in an editor.
- SQL stands for Structured Query Language which allows you to prepare SQL statements.

- SQL is a standard proposed by ANSI (American National Standard Institute).
- There are many vendors who provide databases following RDBMS standards but there are slight variations.
- So when you are working with any particular database you need to check the database manual.
- Some of the databases which follow RDBMS standard are Oracle, MySQL, DB2, etc.

Q What is the difference between DBMS and RDBMS?

A DBMS is a software management system to maintain database. RDBMS is an implementation of DBMS.

Q What is SQL?

A SQL stands for Structured Query Language and is used to communicate with database. SQL is a standard proposed by ANSI.

ORDBMS (Object Relational Database Management System)

- An **object relational database management system (RDBMS)** is also a database management system (DBMS).
- ORDBMS also has the same features of RDBMS.
- ORDBMS database is used to bridge the gap between Relational World and Object Oriented Programming language like Java, C++, and VB etc. We can map directly any language object with the tables of relational database.
- In java applications, we very rarely use ORDBMS support.
- In JAVA, application developers generally use RDBMS and the support of ORM (Object Relational Mapping) technologies like Hibernate, Ibatis, Toplink, etc.
- Today almost all databases like Oracle, MySQL, DB2, etc are ORDBMS but we rarely use the object oriented support. We use only the RDBMS support.

Syntax for creating table:-

```
CREATE TABLE table-name
(
    column-name1 data-type(size) ,
    column-name2 data-type(size),
    column-name3 data-type(size)
);
```

Eg:-

```
CREATE TABLE EMPLOYEE  
(  
    EMPLOYEE_ID NUMBER(5),  
    NAME VARCHAR2(30),  
    SALARY NUMBER(6)  
);
```

Following are the different data types in Oracle which are used extensively.

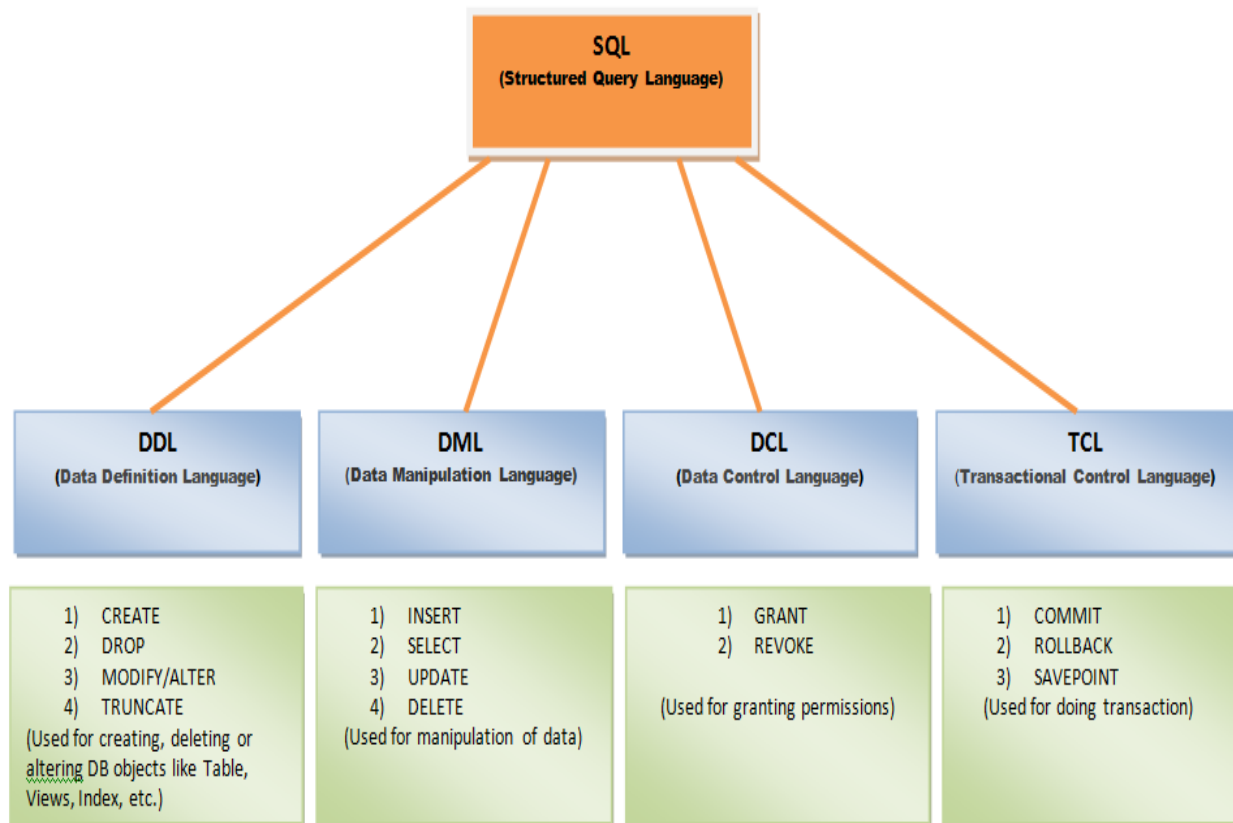
- 1) CHAR: - can store string value but will store blank spaces when the value is short.
- 2) VARCHAR2:- can store string values.
- 3) NUMBER: - can store numeric values.
- 4) DATE: - can store date value.
- 5) BLOB (Binary Large Object):- can store images, videos, sound clips etc.

There are many operators which can be used with WHERE clause:-

- 1) Comparison and relational operators : = , < , <= , > , >= , < > (used for comparing single values)
- 2) Range test operator : BETWEEN
- 3) Set operator: IN ((used for comparing multiple or set of values. We can give multiple values using "IN" as compared to "=" where we can give only one value)
- 4) Pattern matching operator: LIKE (% , _)
% replaces multiple character i.e. zero or more than 1 character
_ replaces single character.
- 5) Null value test operator : IS NULL
- 6) Logical operator: AND, OR NOT

Following are the different data types in MySQL database which are used extensively.

- 1) CHAR:- can store string values
- 2) VARCHAR :- can store string values
- 3) INT:- can store numeric values.
- 4) DATE:- can store date value.
- 5) BLOB (Binary Large Object):- can store images, videos, sound clips etc.



CONSTRAINTS:-

There are 6 types of constraints.

1. NOT NULL
2. UNIQUE
3. CHECK
4. DEFALUT
5. PRIMARY KEY
6. FOREIGN KEY

1. NOT NULL: modifier prevents NULL values from appearing in the column. Null is not equivalent to zero or space or any other value. If a column has a null value it means value has not been provided in the column.
2. UNIQUE: To avoid duplicate values to be added to a table, UNIQUE constraint is used on the column. According to the ANSI standards SQL:92, SQL:1999, and SQL:2003, a UNIQUE constraint should disallow duplicate non-NULL values, but allow multiple NULL values.
3. CHECK: This constraint enforces some business rules on specified columns.
4. DEFAULT: The ANSI/ISO standard allows defining default values that columns should have. Usually, if no value is supplied for a column, then it is assigned a NULL value. The DEFAULT modifier overrides this and assigns the default value.
5. PRIMARY KEY: It enforces both NOT NULL & UNIQUE constraints. This constraint also helps to relate with other tables. Each table can have only one primary key. There are two types of primary key. One is simple primary key and the other is composite primary key.
6. FOREIGN KEY: A foreign key in one table points to a primary key in another table. It is used to establish relationship between two tables. When there are multiple tables in a database the tables must be related to each other. For a column to be defined as a Foreign Key, it should be defined as a Primary Key in the table to which it is referring.

Note: Usually it is a convention to add constraints after creating the table by using ALTER statement. This helps to give a custom name to the PRIMARY KEY; through this name the KEY can be modified/deleted. Otherwise the SQL engine gives an internal name like – “SYS_C000308”.

PRIMARY KEY AND FOREIGN KEY

- Basic rule of database is to avoid null value and redundant value.
- Both PK and FK are generally used if you have 1 to M relationship between entities.
For eg: EMPLOYEE --> EMPLOYEE_PHONE, here 1 employee can have any number of phones (ranging from 0 to N). In this case, if you try to capture the details in the same table you may end up putting nulls to some columns or the other way round, you may not be able to capture the excess data.
- Also maintaining details in a single table will create some problems like:-
 - 1) Data redundancy or duplicity of data.
 - 2) When there are too many columns search operations becomes slow.
 - 3) There will be limitation of columns.
- Advantage of using primary key:-
 - 1) It will not allow invalid data in the child table.
 - 2) It helps in cascading i.e. we delete a record in the master table then it will automatically be deleted in slave or child table.
- By using PK and FK in tables, INSERT will check for integrity and DELETE will do cascading.
- For splitting you can apply thumb rule
 - a) 1 to 1:- store data in same table.
 - b) 1 to M:- store data in another table.
- When it is moved it becomes important to maintain the data integrity (the master/parent data should be linked to the child/slave table). Otherwise invalid data might be stored in child table.
- For doing this we create PK on the parent table.
- FK is created and linked backed to the parent table.
- By doing so:-
 - 1) When inserting a child entry it checks if the associated entry is already present in the parent table or not.
 - 2) While deleting corresponding association is checked between the parent and child and if the parent is deleted automatically child is deleted.
- Usually when tables are split the one side becomes the master or parent and the many side becomes the slave or child. By following the thumb rule the master or one side will have PK and many or slave side will have the FK.
- For every PK in every table we use sequence generator be it static tables or dynamic tables.
Eg:-
EMP_SEQ.NEXTVAL will be for inserting PK.
EMP_SEQ.CURRVAL will be for inserting FK.

SEQUENCE:-

A sequence is an object in Oracle that is used to generate a number sequence. This can be useful when you need to create a unique number to act as a primary key.

NORMALIZATION:

- Normalization is process where large tables are divided into smaller tables and these smaller tables are related with each other. This is mainly done to avoid redundant values (duplicates).
- Having redundant values can cause two major problems
 - 1) Data manipulation (CUD operation) becomes complicated.
 - 2) It occupies more storage space.
- If normalization is done future changes to the data in the database will be easy and hence maintenance reduces. It also occupies less storage space.
- Normalization is mainly done while storing values into database.
- RDBMS databases support normalization.

DENORMALIZATION:

- Denormalization means allowing redundancy in a table.
- It is achieved by storing all the data into a single table.
- The main benefit of denormalization is to improve the read performance of the database or when lot of read queries will be written.
- This is mainly done in data warehousing when creating reports or statistics.

SQL FUNCTIONS :-

Following are the different types of SQL functions:-

1. Conversion functions
2. String functions
3. Mathematical functions.
4. Aggregate functions.

CONVERSION FUNCTIONS	STRING FUNCTIONS	MATHEMATICAL FUNCTIONS	AGGREGATE FUNCTIONS
1) TO_DATE()	1) LENGTH()	1) ABS()	1) MAX()
2) TO_CHAR()	2) SUBSTR()	2) SQRT()	2) MIN()
3) TO_NUMBER()	3) UPPER()	3) SIN()	3) SUM()
4) NVL()	4) LOWER()	4) COS()	4) AVG()
	5) INITCAP()	5) TAN()	5) COUNT()
	6) LTRIM()	6) CEIL()	
	7) RTRIM()	7) FLOOR()	
	8) LPAD()	8) ROUND()	
	9) RPAD()		

1. Conversion functions

Following are the different types of conversion functions:-

1) TO_DATE() :- (string to date)

- TO_DATE function actually converts a string to date format.
- This function is generally used when you are doing CUD operations.

```
INSERT INTO EMP_DATE(NAME,DOB) VALUES ('AAA',TO_DATE('2011/09/23','YYYY/MM/DD'));
```

- This function will take both the values and will do the conversion and then store in database.
- SYSDATE is a function that returns date and time.

2) TO_CHAR() :- (date to string)

- For displaying we convert date to string. Then use TO_CHAR function.
- It is generally used when you are doing read operation (SELECT).
- While storing date Oracle stores in its own numerical format but while displaying it will always display in the default format of DD-MON-YY format.
- If you want any other format other than this, then use TO_CHAR function.

3) TO_NUMBER() :- converts string to number.

- DUAL is a SYNONYM for a table.
- SYNONYM is a DB object.
- DUAL is a table with single row and single column with the name DUMMY.

DUAL (Table Name)

DUMMY (col name)

4) NVL function:-

- It is used for avoiding null values in a column while doing read operation (SELECT).

AGGREGATE FUNCTIONS OR GROUPING FUNCTIONS:-

- Aggregate functions can be used only with SELECT and HAVING clause.
 - The argument used with an aggregate function is always a column name.
 - Aggregate functions can be applied only on groups.
 - Groups can be formed by using GROUP BY clause.
 - If there are no groups, then the entire table will be considered as one group.
 - When an aggregate function is used with a SELECT clause, then no other column can be used with the SELECT clause. However if a GROUP BY clause is used then the grouping "attribute" name (*attribute name means column name*) can be used with SELECT clause.
 - Aggregate functions always return a single value. If there are many groups then one value will be returned for every group.
 - Following are the list of aggregate functions.
 - 1) MAX() :- returns the highest value in the specified column.
 - 2) MIN():- returns the lowest value in the specified column.
 - 3) SUM():- returns the sum of all values in the specified column.
 - 4) AVG():- returns the average value in the specified column.
 - 5) COUNT():- returns the number of rows.
- Note:-All aggregate functions except COUNT(*) ignore null values. If you want to substitute null with a value then use NVL() function.

SQL ALIAS:-

- We can use alias names for columns and tables.
- Alias is used to make the column names in the results more readable.
- Alias is used generally when
 - 1) The column names are big and not readable.
 - 2) Functions are use in a SELECT query.
 - 3) When there are more than two tables used in a SELECT query.
- An alias name is a temporary name and it does not make any changes to the table names or column names in the database.

STRUCTURE OF SQL QUERY

STRUCTURE OF SQL QUERY

- 6 SELECT :-** is used to select columns from tables. We can also use SQL functions along with SELECT.
- 1 FROM :-** is used to select tables or view.
- 2 ON:-** is used to provide condition to join tables.
- 3 WHERE :-** is used to select rows from tables by providing conditions using column name.
- 4 GROUP BY :-** is used to form groups. You have to provide the attribute name (column name) on which grouping has to be done.
- 5 HAVING:-** is used to select groups by providing condition using aggregate functions.
- 7 ORDER BY:-** is used for sorting the results either by ascending or descending. Default is ascending.

Following are the different clauses which can be used in the select query:

1. SELECT:- clause is used for selecting the columns from tables. We can also use SQL functions along with SELECT clause. When you use an aggregate function no other column can be used for select operation. However if you use GROUP BY clause you can use only the grouping attribute column name with SELECT clause.
2. FROM:- clause is used for selecting the tables. We can also use FROM clause for selecting the views.
3. WHERE :- clause is used for selecting the rows from tables by giving conditions using operators.
4. GROUP BY:- clause is used to form groups.
 - Always aggregate functions will be applied on groups and to create groups we use GROUP BY clause.
 - When you are using GROUP BY clause no other column can be used for select operation along with the SELECT clause apart from the grouping attribute column name. However you can use only aggregate function along with the SELECT clause.
5. HAVING :- clause is used to select groups by giving conditions using aggregate function with operators.
 - If you want to use HAVING clause, GROUP BY is mandatory because HAVING clause will use aggregate function and to use aggregate function grouping is required.

- But when you are using GROUP BY, HAVING is optional.
- 6. ORDER BY : clause is used for ordering the output of query either by ascending or descending.
- 7. ON : is used for joining tables by providing conditions.

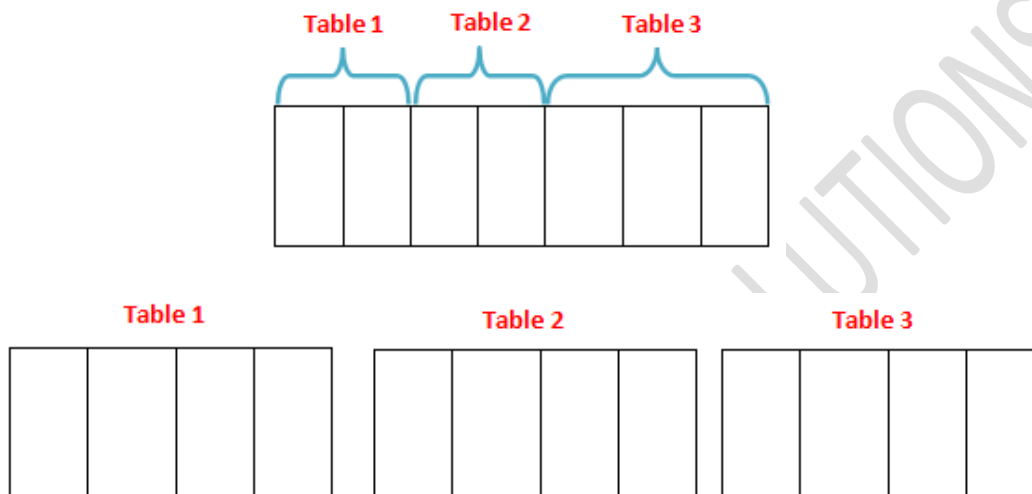
Q What is the difference between HAVING and WHERE clause?

A Following are the differences:

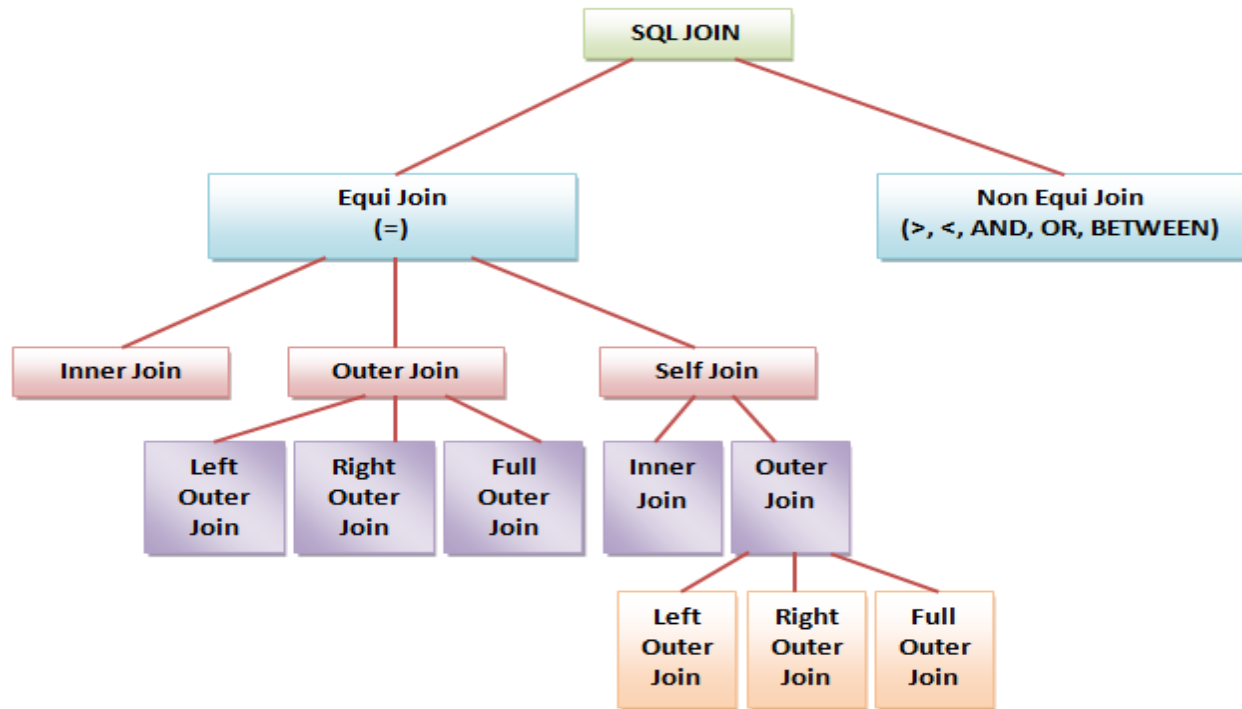
WHERE	HAVING
WHERE is used to provide condition to select rows.	HAVING is used to provide condition to select groups.
WHERE condition is given by using column names.	HAVING condition is given by using aggregate function.
WHERE can be used with SELECT, UPDATE and DELETE query.	HAVING can be used with only SELECT query.
We cannot use aggregate function with WHERE clause.	We can use aggregate function with HAVING clause.

SQL JOINS

- Creating n tables to a single table is called JOIN.
- SQL join is used for retrieving data from multiple tables.
- While storing we store in multiple tables.
- While retrieving we get into a single table.



- For joining table's criteria is the data type of columns. The data type of the columns which are used for joining should be of the same type.
- To join tables, you need not have a relationship between PK and FK.
- We can join two unrelated tables also.
- Join can work without PK and FK, but they are needed for inserting and deleting.
- While inserting PK & FK will check for integrity.
- While deleting PK & FK will do cascading.

Different type of joins:-

- When we are joining the tables, we write the condition using many operators.
- When we use “=” (comparison) operator, then it is called as equi join.
- If we use any other operator other than “=” operator like (>, <, AND, OR, BETWEEN) then we call it as non-equi join.

There are three types of equi joins:-

1. Inner join: - In the inner join only the matching records will be selected and joined in the result.
2. Outer join: - In an outer join matching records will be selected plus extra records will be selected.

Outer join is the result of inner join plus extra records.

Outer join = inner join + extra records.

There are three types of outer joins:-

- a) Left outer join:- It will select all matching records plus extra records in the left table.

Left outer join = inner join + extra records in the left table.

- b) Right outer join:- It will select all matching records plus extra records in the right table.

Right outer join = inner join + extra records in right table.

- c) Full outer join:- It will select all matching records plus extra records in both tables.

Full outer join = inner join + extra records in both tables.

3. Self join:- If a table is joined with itself, then it is called as self join.

- Cross join is the Cartesian product of the two tables.

VIEW

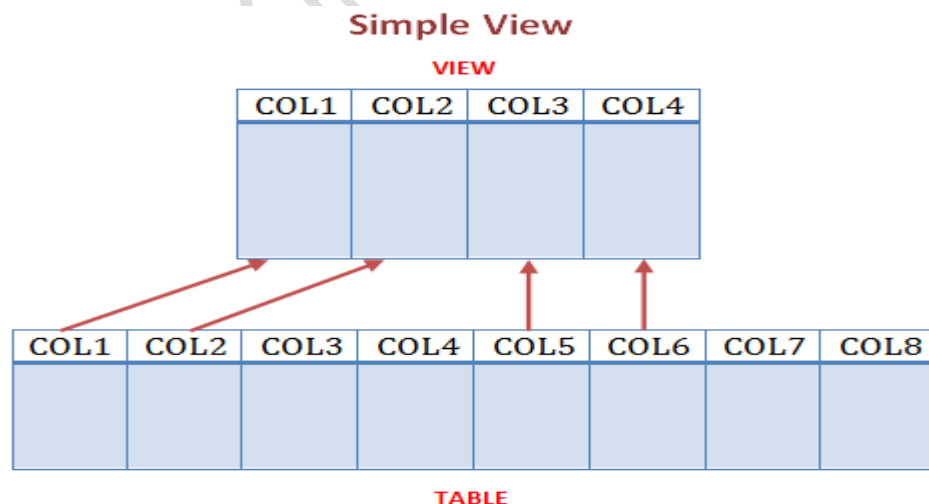
- It is a DB object.
- It is a virtual table.
- It is a logical representation of the data in the table.

Advantages:-

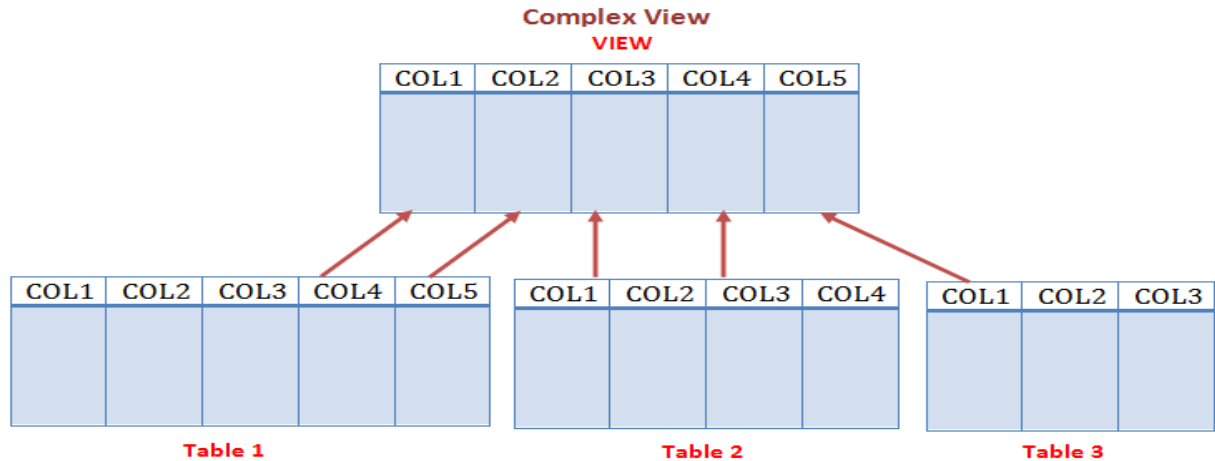
- 1) It helps in providing security i.e. it helps to hide columns having sensitive info in the underlying table.
- 2) Reduces writing complicated join queries.
- 3) Reduces data redundancy.

A view can be classified into 2 types.

1. Simple view: - is created from a single table.



2. Complex view: - is created from multiple tables.



- Any changes in table will affect view and any changes in view will also affect table.
- Always use view for Read operation.
- Never use view for CUD operations. Always do CUD operations directly on the table.

INDEX

- It is a DB object.
- Indexes allow the database to find data fast, without reading the whole table.

Advantages:-

1. Faster searches and thereby it gives improved performance.

- Indexes should be created only on columns where lots of read operations are happening.
- But never create index for columns where lot of CUD operations are happening.
- When a Primary Key or Unique Key is added on a column, that column is indexed implicitly.
- Creating index on these 2 types of columns will give error.
- Any column apart from Primary Key and Unique Key needs to be indexed explicitly.
- The users cannot see the index; it is just used to speed up searches/queries.

SYNONYM

- SYNONYM is a DB object.
- DUAL is one of the examples for SYNONYM.
- DUAL is a SYNONYM for a table.
- DUAL is a table with single row and single column with the name DUMMY.

DUAL (Table Name)

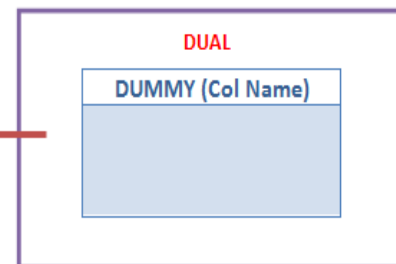
DUMMY (Col Name)

- SYSTEM has granted permission to access this table DUAL which is present in SYSTEM schema to all users or schemas.

RAVIORACLE (Schema)



SYSTEM(Schema)



Can access

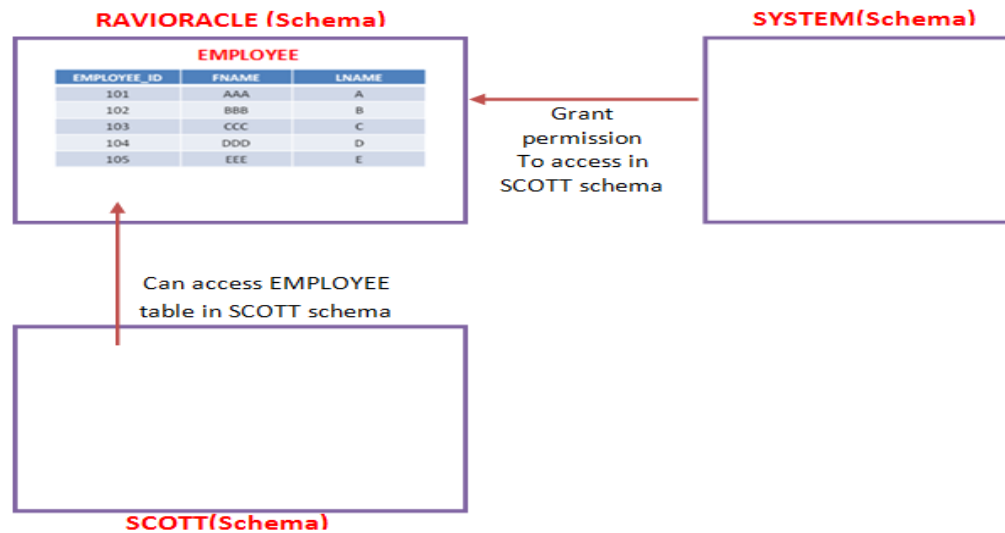


SELECT * FROM SYSTEM.DUAL

- This is the actual way we have to access the table in another schema apart from SYSTEM.
- But we do it like this.

SELECT * FROM DUAL

- SYNONYM is actually created to avoid using full schema name in another schema to access a DB object.
- A SYNONYM is an alternative name for objects such as tables, views, sequences, stored procedures and other database objects.

Steps for creating SYNONYM.

1. Create a table called EMPLOYEE in schema RAVORACLE by logging in as RAVORACLE.
2. Log into SYSTEM schema and grant permission to SCOTT to access the table EMPLOYEE table present in RAVORACLE.

```
GRANT SELECT, INSERT, UPDATE, DELETE ON RAVORACLE.EMPLOYEE TO SCOTT;
```

3. Login to SCOTT.

```
SELECT * FROM EMPLOYEE;
```

This will not work because you have to access with schema name.

```
SELECT * FROM RAVORACLE.EMPLOYEE;
```

This will work.

4. Create the synonym in SCOTT schema.

```
CREATE OR REPLACE PUBLIC SYNONYM EMPLOYEE_SYN FOR RAVORACLE.EMPLOYEE;
```

5. Now we can access it in SCOTT in the following way.

```
SELECT * FROM EMPLOYEE_SYN;
```

(This query will access the table via the synonym).