Oracle 11 g (SQL)

Database Introduction

DAY-1

1. Database Introduction
    - Data
    - Database
    - Metadata
    - Database Management System
2. Data types in Oracle
3. SQL statements
    DDL, DML, DCL, TCL, Data retrieval
4. Create table statement
5. Populating data into tables
    - Insert statement
6. Data retrieval using SELECT statement
7. Arithmetic operators, operator precedence
    - Addition (+)
    - subtraction (-)
    - multiply (*)
    - divide (/)
8. Handling null values
9. Working with aliases
    - column aliases
    - table aliases

10. Concatenation operator (||)
11. Suppressing duplicate rows
    - distinct keyword
12. Filtering of records
    - Where clause
13. Logical operators
    - AND, OR, NOT
14. Comparison operators
    =, <>/!=/^=, <, >, <=, >=
15. SQL * Plus operators
    - BETWEEN … AND …
    - IN
    - IS NULL
    - LIKE
16. Ordering information
    ORDER BY
17. SQL functions
18. Working with dates

Data

The contents of a table are stored in rows are referred as “Data”

Metadata:

It is data which describes the properties of end users data.
Metadata properties can include the **information such as:**

Table name,
Table owner,
details about the columns (Values allowed, length or size) and the
physical storage size on disk.

Data Base

A **database** is an organized collection of data. Since the data in a database is organized it makes data management easy.

Database Management System (DBMS)

Database management system (DBMS) is a Software (collection of programs) which enables user to create, access database, provide controlled access to user databases and manipulate data .

Data Types in Oracle:

when creating tables, each column must be assigned a data type, which determines the nature of the values that can be inserted into the columns.

NUMBER data type

- It Stores zero, positive, and negative fixed and floating-point numbers
- **The general declaration is: NUMBER(P,S)**
- P → it specifies the precision, i.e the total no of digits (1 to 38).
- S → it specifies the scale, the number of digits to the right of decimal point, can range from -84 to 127

VARCHAR2 data type

- Specifies the Variable length Character string.
- Minimum size is 1 byte maximum size is 4000 bytes.
- It occupies only that space for which the data is supplied.

CHAR data type

- It specifies Fixed length Character string.
- The size should be specified.
- If the data is less than the original specified size, blank pads are applied.
- The default length is 1 byte and maximum is 2000 bytes

DATE data type

- It is used to store date and time information.
- The information revealed by data is :
  century, year, month, date, hour, minute, second.
- Default Date format in oracle is DD-MON-YY.
- The date Range provided by oracle is January 1, 4712 BC to December 31, 9999 AD.

- NVARCHAR2
- RAW
- FLOAT
- INTEGER
- TIMESTAMP
- CLOB
- NCLOB
- BLOB
- BFILE
- LONG
- LONG RAW
- ROWID

SQL Statements

DDL - Data Definition Language

CREATE : to create a new data structure.
ALTER : to change an existing data structure.
DROP : to remove an entire data structure.
TRUNCATE : to remove all rows permanently from table
RENAME : to rename existing table

DML - Data Manipulation Language

INSERT : to add records into the table
UPDATE : to change column value in the table
DELETE : to remove rows from the table

Data Retrieval Language

SELECT : used to retrieve the data present in the database tables

DCL - Data Control Language

GRANT :Allow access privileges to users
REVOKE :Revoke or cancel access privileges

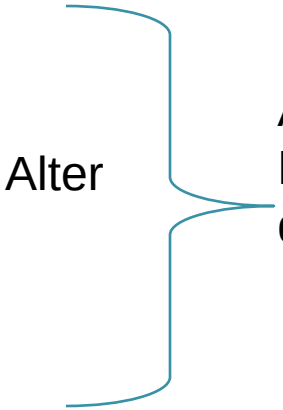
TCL-Transaction Control Language.

COMMIT :Save or enable DML changes to the database.
ROLLBACK : To undo DML changes till in a transaction
SAVEPOINT: To divide a transaction

DDL - Data Definition Language

CREATE : to create a new data structure.
ALTER : to change an existing data structure.
DROP : to remove an entire data structure.
TRUNCATE : to remove all rows permanently from table
RENAME : to rename existing table

The ALTER statement is used to add, drop and modify columns in an existing table.



To add a column in a table:
ALTER TABLE table_name
ADD (column_name datatype)

Note: if the table contains records, before the Column is added the new columns
NULL values.

To drop a column from a table

ALTER TABLE table_name
DROP COLUMN column_name

Note: only one column can be droped at a time
The column may or maynot contain data.
Once a column is droped it cannot be recovered
Modify column:
Alter table <table_name>
Modify (column datatype)

Description	**SYNTAX:**	**Note:**
Adding a column to a table	ALTER TABLE table_name ADD (column_name datatype)	if the table contains records, before the Column is added the new column contains NULL values.
Dropping a column	ALTER TABLE table_name DROP COLUMN column_name	❑only one column can be droped at a time. ❑The column may or maynot contain data. ❑Once a column is droped it cannot be recovered.
Modify column:	Alter table <table_name> Modify (column datatype)	❑We can decrease the width of a column if contains only Null values and if the table has no data

- We can chage the datatype if the column contains NULL's

Drop:

It removes the definition of the oracle table(including data) and associated INDEXES

Syntax: drop table <table name>

Any views and synonyms will remain nut are kept in invalid state.

Drop table statement once executed is irreversible.

Changing the name of an object:

The rename command can be used to change the name of a
Table
View
Sequence
Synonym

Syntax:
Rename <oldname> to <newname>

Truncation a table:

It is used to remove all rows from a table and to release the storage space

Used by the specific table.

Syntax:

Truncate table <table_name>

Create table statement :

Syntax:

CREATE TABLE <table_name>
(
column_name 1<Data type> (width),
column_name 2<Data type> (width),
.....
;;;;;;;;;
column_name N<Data type> (width)

)

Note: all data types may not have the width property.
no two columns in the same table can have the same name

Filter data:

```
select ename, sal, deptno, job from emp
where deptno = 10 and
job='MANAGER' or
job='PRESIDENT' or
job = 'CLERK'
and
sal >=1300 and sal <=5000
/
```

SQL * PLUS OPERATORS:

1. Between … and … operator:

This operator is used to display rows based on range of values
The declared range is inclusive
The lower limit should be declared first
The negation of this operator is NOT between … and …

2. IN operator:

The operator is used for test for values in a specified list.
The operator can be used upon any data type.
The negation of the operator is NOT IN.

3. Is null operator:

Test for null values
It is the only operator that can be used to test for NULL's.
The negation is IS NOT NULL

Comparison operators:

The comparison operators are used in such conditions that compare One expression to another expression.

Different comparison operators are:

= → equality operator
<>,!=, ^= → not equality operator
> →Greater than operator
< → less than operator
<= , > = → less than or equal to operator , greater than or equal to

Applying logical operators to filters:

The logical operators combine the results of two component conditions To produce a single result.

Following are the logical operators in oracle:

And:
It returns true if both or all component conditions are TRUE.
It returns false if either is False, else return unknown.
Or:
It returns TRUE if either of the component condition is TRUE.

Not:
It returns true if the following condition is False.
It returns False if the following condition is TRUE.

ALTER command:

ADD
MODIFY
DROP

- ALTER TABLE <table_name> ADD (Columnname Datatype,--,--)
- alter table <table_name> modify (ENAME varchar2(20))
- alter table <table_name> drop column column_name

DROP Command:

It removes the definition Of the table, data and Associated indexes.

SYNTAX:

- **Drop table table_name**

CASCADE CONSTRAINTS

> show recyclebin;

TRUNCATE

It is used to remove all rows from a table and To release the storage space used by the Specific Table.

Note: the action of the TRUNCATE command Cannot be rolled back.

Syntax:Truncate table table_name;

RENAME:

It is used to change the name of a table.

Syntax:
Rename <oldtable_Name> to
<New_Table_name> ;

DML

DML - Data Manipulation Language
- We need to commit explicitly.

Commands :-

INSERT : to add records into the table
UPDATE : to change column value in the table
DELETE : to remove rows from the table

Example

- SQL> insert into Dept values (1,'Ramu','a');
- SQL> insert into Dept (deptno,dname) values (2,'Gopi');
- SQL> insert into Dept values(&deptno,'&Dname','&loc');

  Enter the Value for Deptno: 3

  Enter the Value for Dname: Ravi

  Enter the Value for Dname: Hyd
- SQL> insert into Dept (deptno,dname) values (&deptno,'&Dname');

UPDATE Command:

The UPDATE statement is used to change the existing values in a table or in the Base table of view.

Syntax:
UPDATE <table_name> SET <specification> WHERE clause.

DELETE Statement:

It is used to remove rows from table.

Syntax: DELETE [from] <table_name>
[where condition];

DCL

DCL - Data Control Language

- Used to Control the information between Users.

GRANT :Allow access privileges to users

REVOKE :Revoke or cancel access privileges

DCL

- At a time we will give permission on One Table.
- If parent lost its privileges then Child will also loss the Privileges.
- Parent User can Revoke the Privileges from Childs and Grand Childs.

SQL Functions

SQRT

- SQL> select SQRT(4) from dual;

```
SQRT(4)
----------
         2
```

- SQL> select SQRT(2) from dual;

```
SQRT(2)
----------
1.41421356
```

ROUND

- **SQL> SELECT ROUND(19) FROM DUAL;**

```
ROUND(19)
----------
        19
```

- **SQL> SELECT ROUND(19.4) FROM DUAL;**

```
ROUND(19.4)
-----------
         19
```

- **SQL> SELECT ROUND(19.5) FROM DUAL;**

```
ROUND(19.5)
-----------
         20
```

ROUND

- Conti….
- SQL> SELECT ROUND(19.254,2) FROM DUAL;

```
ROUND(19.254,2)
---------------
        19.25
```

- SQL> SELECT ROUND(19.255,2) FROM DUAL;

```
ROUND(19.255,2)
---------------
        19.26
```

- SQL> SELECT ROUND(19.256,2) FROM DUAL;

```
ROUND(19.256,2)
---------------
        19.26
```

TRUNC

- SQL> SELECT TRUNC(19) FROM DUAL;

```
 TRUNC(19)
----------
        19
```

- SQL> SELECT TRUNC(19.4) FROM DUAL;

```
TRUNC(19.4)
-----------
         19
```

- SQL> SELECT TRUNC(19.5) FROM DUAL;

```
TRUNC(19.5)
-----------
         19
```

Conti…

TRUNC

Conti…

- SQL> SELECT TRUNC(19.124,2) FROM DUAL;

```
TRUNC(19.124,2)
---------------
          19.12
```

- SQL> SELECT TRUNC(19.125,2) FROM DUAL;

```
TRUNC(19.125,2)
---------------
          19.12
```

- SQL> SELECT TRUNC(19.126,2) FROM DUAL;

```
TRUNC(19.126,2)
---------------
          19.12
```

Character Functions

1. UPPER ( )
2. LOWER ( )
3. INITCAP ( )
4. LENGTH ( )
5. REVERSE ( )
6. CONCAT ( )
7. LPAD( )
8. RPAD( )
9. TRIM( )
10. LTRIM( )
11. RTRIM ( )
12. ASCII( )
13. CHR ( )
14. VSIZE ( )
15. SUBSTR( )
16. INSTR ( )

Department Table

DEPT NO	DNAME	LOC
10	ACCOUNTING	NEWYORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

Employee Table

ENO	ENAME	JOB	MGR	H-DATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7788	SCOTT	ANALYST	7566	19-APR-87	3000		20
7839	KING	PRESIDENT		17-NOV-81	5000		10
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7876	ADAMS	CLERK	7788	23-MAY-87	1100		20
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7902	FORD	ANALYST	7566	03-DEC-81	3000		20
7934	MILLER	CLERK	7782	23-JAN-82	1300		10

Examples

- SQL> SELECT ENAME,LOWER(ENAME) LOWER,
  UPPER(ENAME) UPPER,
  INITCAP(ENAME) INIT
  FROM EMP;

ENAME	*LOWER*	*UPPER*	*INIT*
SMITH	smith	SMITH	Smith
ALLEN	allen	ALLEN	Allen
WARD	ward	WARD	Ward
JONES	jones	JONES	Jones
MARTIN	martin	MARTIN	Martin
BLAKE	blake	BLAKE	Blake
CLARK	clark	CLARK	Clark
SCOTT	scott	SCOTT	Scott
KING	king	KING	King
TURNER	turner	TURNER	Turner
ADAMS	adams	ADAMS	Adams
JAMES	james	JAMES	James
FORD	ford	FORD	Ford
MILLER	miller	MILLER	Miller

REVERSE

- SQL> SELECT ENAME, REVERSE(ENAME) FROM EMP;

```
ENAME      REVERSE(EN
----------  ----------
SMITH      HTIMS
ALLEN      NELLA
WARD       DRAW
JONES      SENOJ
MARTIN     NITRAM
BLAKE      EKALB
CLARK      KRALC
SCOTT      TTOCS
KING        GNIK
TURNER     RENRUT
ADAMS      SMADA
JAMES      SEMAJ
FORD       DROF
MILLER     RELLIM
```

```
SQL> SELECT ENAME, LENGTH(ENAME)LEN_ENAME,
                DEPTNO, LENGTH(DEPTNO) LEN_DNO
                                    FROM EMP;
```

ENAME	LEN_ENAME	DEPTNO	LEN_DNO
SMITH	5	20	2
ALLEN	5	30	2
WARD	4	30	2
JONES	5	20	2
MARTIN	6	30	2
BLAKE	5	30	2
CLARK	5	10	2
SCOTT	5	20	2
KING	4	10	2
TURNER	6	30	2
ADAMS	5	20	2
JAMES	5	30	2
FORD	4	20	2
MILLER	6	10	2

CONCAT

SQL> select ename,sal, concat(ename,sal) CONC from emp;

ENAME	*SAL*	*CONC*
SMITH	800	SMITH800
ALLEN	1600	ALLEN1600
WARD	1250	WARD1250
JONES	2975	JONES2975
MARTIN	1250	MARTIN1250
BLAKE	2850	BLAKE2850
CLARK	2450	CLARK2450
SCOTT	3000	SCOTT3000
KING	5000	KING5000
TURNER	1500	TURNER1500
ADAMS	1100	ADAMS1100
JAMES	950	JAMES950
FORD	3000	FORD3000
MILLER	1300	MILLER1300

PADDING

```
SQL> SELECT ENAME,LPAD (ENAME,15,'@') L_PAD ,
                  RPAD (ENAME,15,'@') R_PAD
                                      FROM EMP;
```

ENAME	L_PAD	R_PAD
SMITH	@@@@@@@@@@SMITH	SMITH@@@@@@@@@@
ALLEN	@@@@@@@@@@ALLEN	ALLEN@@@@@@@@@@
WARD	@@@@@@@@@@WARD	WARD@@@@@@@@@@
JONES	@@@@@@@@@@JONES	JONES@@@@@@@@@@
MARTIN	@@@@@@@@@MARTIN	MARTIN@@@@@@@@@
BLAKE	@@@@@@@@@@BLAKE	BLAKE@@@@@@@@@@
CLARK	@@@@@@@@@@CLARK	CLARK@@@@@@@@@@
SCOTT	@@@@@@@@@@SCOTT	SCOTT@@@@@@@@@@
KING	@@@@@@@@@@@KING	KING@@@@@@@@@@@
TURNER	@@@@@@@@@TURNER	TURNER@@@@@@@@@
ADAMS	@@@@@@@@@@ADAMS	ADAMS@@@@@@@@@@
JAMES	@@@@@@@@@@JAMES	JAMES@@@@@@@@@@

TRIM

SQL> select ename, trim(ename) from emp;

ENAME	*TRIM(ENAME*
SMITH	SMITH
ALLEN	ALLEN
WARD	WARD
JONES	JONES
MARTIN	MARTIN
BLAKE	BLAKE
CLARK	CLARK
SCOTT	SCOTT
KING	KING
TURNER	TURNER
ADAMS	ADAMS
JAMES	JAMES
FORD	FORD
MILLER	MILLER

Left and Right TRIM

Left Trim:

```
SQL> select ename,
    ltrim(ename) from emp;
```

```
ENAME      LTRIM(ENAM
----------      ----------
SMITH      SMITH
ALLEN      ALLEN
WARD       WARD
JONES      JONES
MARTIN     MARTIN
BLAKE      BLAKE
CLARK      CLARK
SCOTT      SCOTT
KING       KING
TURNER     TURNER
ADAMS      ADAMS
```

Right Trim:

```
SQL> select
    ename,rtrim(ename) from
    emp;
```

```
ENAME      RTRIM(ENAM
---------- -    ---------
SMITH      SMITH
ALLEN      ALLEN
WARD       WARD
JONES      JONES
MARTIN     MARTIN
BLAKE      BLAKE
CLARK      CLARK
SCOTT      SCOTT
KING       KING
TURNER     TURNER
ADAMS      ADAMS
```

Left and Right TRIM

Left trim:

- **SQL> select ename ,ltrim(ename,'S') from emp;**

```
ENAME      LTRIM(ENAM
---------- -  ---------
SMITH      MITH
ALLEN      ALLEN
WARD       WARD
JONES      JONES
MARTIN     MARTIN
BLAKE      BLAKE
CLARK      CLARK
SCOTT      COTT
KING       KING
TURNER     TURNER
ADAMS      ADAMS
JAMES      JAMES
FORD       FORD
MILLER     MILLER
```

Right Trim:

SQL> select ename ,rtrim(ename,'R') from emp;

```
ENAME      RTRIM(ENAM
---------- ----------
SMITH      SMITH
ALLEN      ALLEN
WARD       WARD
JONES      JONES
MARTIN     MARTIN
BLAKE      BLAKE
CLARK      CLARK
SCOTT      SCOTT
KING       KING
TURNER     TURNE
ADAMS      ADAMS
JAMES      JAMES
FORD       FORD
```

```
SQL> select ASCII('a') from dual;

ASCII('A')
----------
        97
SQL> select ASCII('A') from dual;

ASCII('A')
----------
        65
SQL> select CHR(97) from dual;

C
-
a

SQL> select CHR(65) from dual;

C
-
A
```

```
SQL> desc emp;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMPNO                                     NOT NULL NUMBER(4)
 ENAME                                              VARCHAR2(10)
 JOB                                                VARCHAR2(9)
 MGR                                                NUMBER(4)
 HIREDATE                                           DATE
 SAL                                                NUMBER(7,2)
 COMM                                               NUMBER(7,2)
 DEPTNO                                             NUMBER(2)
```

SUBSTR

- SQL> select SUBSTR('Rajesh',1,3) from dual;

```
SUB
---
Raj
```

- SQL> select SUBSTR('Rajesh',3) from dual;

```
SUBS
----
Jesh
```

- SQL> select SUBSTR('Rajesh',1) from dual;

```
SUBSTR
------
Rajesh
```

SUBSTR Conti….

```
SQL> select ename,substr(ename,1,4) from emp;
```

```
ENAME      SUBS
---------- ----
SMITH      SMIT
ALLEN      ALLE
WARD       WARD
JONES      JONE
MARTIN     MART
BLAKE      BLAK
CLARK      CLAR
SCOTT      SCOT
KING       KING
TURNER     TURN
ADAMS      ADAM
JAMES      JAME
FORD       FORD
MILLER     MILL
```

```
SQL> select instr('Database','a',1,1) from dual;

INSTR('DATABASE','A',1,1)
-------------------------
                        2

SQL> select instr('Database','A',1,1) from dual;

INSTR('DATABASE','A',1,1)
-------------------------
                        0

SQL> select instr('Database','a',1,2) from dual;

INSTR('DATABASE','A',1,2)
-------------------------
                        4
```

```
SQL> select instr('Database','a',1) from dual;

INSTR('DATABASE','A',1)
-----------------------
                      2
SQL>  select instr('Database','a',2) from dual;
INSTR('DATABASE','A',2)
-----------------------
                      2
SQL> select instr('Database','a',3,1) from dual;
INSTR('DATABASE','A',3,1)
-------------------------
                        4
SQL> select instr('Database','a',3,2) from dual;
INSTR('DATABASE','A',3,2)
-------------------------
                        6
```

Date Functions

Function	Description
MONTHS_BETWEEN	Number of months between two dates
ADD_MONTHS	Add calendar months to date
NEXT_DAY	Next day of the date specified
LAST_DAY	Last day of the month

Date Functions

2) Months_between(D1 , D2)

3) Add_Months(D ,+/- N)

3) Last_day (D)

4) Next_day (D ,'Day')

SYSDATE

```
SQL> SELECT SYSDATE FROM DUAL;

SYSDATE
---------
14-JUL-07
```

Adding DAYS and MONTHS

ADDING DAYS

```
SQL> SELECT SYSDATE FROM
    DUAL;

SYSDATE
---------
14-JUL-07

SQL> SELECT SYSDATE + 10 FROM
    DUAL;

SYSDATE+1
---------
24-JUL-07

SQL> SELECT SYSDATE - 10 FROM
    DUAL;

SYSDATE-1
---------
04-JUL-07
```

ADD_MONTHS

```
SQL> SELECT ADD_MONTHS(SYSDATE,2)
    FROM DUAL;

ADD_MONTH
---------
14-SEP-07
SQL>  SELECT ADD_MONTHS(SYSDATE,-2)
    FROM DUAL;

ADD_MONTH
---------
14-MAY-07
SQL> SELECT ADD_MONTHS(SYSDATE,1.9)
    FROM DUAL;

ADD_MONTH
---------
14-AUG-07
SQL> SELECT ADD_MONTHS(SYSDATE,0)
    FROM DUAL;
ADD_MONTH
---------
```

LAST_DAY

```
SQL>   SELECT LAST_DAY(SYSDATE) FROM DUAL;

LAST_DAY(
---------
31-JUL-07

SQL> SELECT LAST_DAY(ADD_MONTHS(SYSDATE,1)) FROM DUAL;

LAST_DAY(
---------
31-AUG-07
```

NEXT_DAY

```
SQL> SELECT SYSDATE FROM DUAL;

SYSDATE
---------
14-JUL-07

SQL> SELECT TO_CHAR(SYSDATE,'DAY') FROM DUAL;

TO_CHAR(S
---------
SATURDAY

SQL> SELECT NEXT_DAY(SYSDATE,'MON') FROM DUAL;

NEXT_DAY(
---------
16-JUL-07
```

NEXT_DAY

```
SQL> SELECT NEXT_DAY(SYSDATE,2) FROM DUAL;

NEXT_DAY(
---------
16-JUL-07

SQL> SELECT NEXT_DAY(SYSDATE,1) FROM DUAL;

NEXT_DAY(                    1--SUNDAY
---------                    2-MONDAY
15-JUL-07

SQL>  SELECT NEXT_DAY(SYSDATE,6) FROM DUAL;
NEXT_DAY(
---------
20-JUL-07
```

Aggregate Functions

Aggregate Functions

1. MIN ( )
2. MAX ( )
3. SUM ( )
4. AVG ( )
5. COUNT (* / Col)

Aggregate Functions

```
SQL> select * from emp;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
```

Aggregate Functions

```
SQL> select Min(sal) from emp;

  MIN(SAL)
----------
       800

SQL> Select Max(sal) from emp;

  MAX(SAL)
----------
      5000

SQL> Select Sum(sal) from emp;

  SUM(SAL)
----------
     29025

SQL> Select Sum(sal) from emp where deptno=10;

  SUM(SAL)
----------
      8750
```

Aggregate Functions

```
SQL> Select Avg(Sal) from emp;

  AVG(SAL)
----------
2073.21429

SQL> select Avg(Sal) from emp where deptno=10;

  AVG(SAL)
----------
2916.66667

SQL> select count(*) from emp;

  COUNT(*)
----------
        14

SQL> select count(comm) from emp;

COUNT(COMM)
-----------
         4
```

Aggregate Functions

SQL> SELECT * FROM EMP ORDER BY DEPTNO;

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7782	**CLARK**	**MANAGER**	**7839**	**09-JUN-81**	**2450**		**10**
7839	**KING**	**PRESIDENT**		**17-NOV-81**	**5000**		**10**
7934	**MILLER**	**CLERK**	**7782**	**23-JAN-82**	**1300**		**10**
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7902	FORD	ANALYST	7566	03-DEC-81	3000		20
7876	ADAMS	CLERK	7788	23-MAY-87	1100		20
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7788	SCOTT	ANALYST	7566	19-APR-87	3000		20
7521	**WARD**	**SALESMAN**	**7698**	**22-FEB-81**	**1250**	**500**	**30**
7844	**TURNER**	**SALESMAN**	**7698**	**08-SEP-81**	**1500**	**0**	**30**
7499	**ALLEN**	**SALESMAN**	**7698**	**20-FEB-81**	**1600**	**300**	**30**
7900	**JAMES**	**CLERK**	**7698**	**03-DEC-81**	**950**		**30**
7698	**BLAKE**	**MANAGER**	**7839**	**01-MAY-81**	**2850**		**30**
7654	**MARTIN**	**SALESMAN**	**7698**	**28-SEP-81**	**1250**	**1400**	**30**

Aggregate Functions

SQL> SELECT **DEPTNO , SUM (SAL)** FROM EMP;
SELECT DEPTNO , SUM (SAL) FROM EMP
*

ERROR at line 1:
ORA-00937: not a **single-group group function**

SQL> SELECT DEPTNO , SUM (SAL) FROM EMP **GROUP BY DEPTNO**;

DNO	SUM(SAL)
10	**8750**
20	**10875**
30	**9400**

Aggregate Functions

```
SQL>  SELECT DEPTNO , JOB , SUM (SAL) FROM EMP GROUP BY DEPTNO;
 SELECT DEPTNO , JOB , SUM (SAL) FROM EMP GROUP BY DEPTNO
                *
ERROR at line 1:
ORA-00979: not a GROUP BY expression
```

```
SQL> Select  DEPTNO , JOB , SUM (SAL) From Emp
       GROUP BY DEPTNO,job order by deptno;

    DEPTNO JOB         SUM(SAL)
---------- --------- ----------
        10 CLERK             1300
        10 MANAGER           2450
        10 PRESIDENT         5000
        20 ANALYST           6000
        20 CLERK             1900
        20 MANAGER           2975
        30 CLERK              950
        30 MANAGER           2850
        30 SALESMAN          5600
```

Aggregate Functions

SQL> SELECT DEPTNO , SUM (SAL) FROM EMP **GROUP BY DEPTNO**;

```
    DNO   SUM(SAL)
---------- -------------------- ----------
        10       8750
        20      10875
        30       9400
```

SQL> select DEPTNO , SUM(SAL) FROM EMP
WHERE DEPTNO IN (10,20) GROUP BY DEPTNO;

```
    DEPTNO   SUM(SAL)
---------- ----------
        20      10875
        10       8750
```

SQL> select DEPTNO , SUM(SAL) FROM EMP
GROUP BY DEPTNO
HAVING SUM(SAL) < 10000;

```
    DEPTNO   SUM(SAL)
---------- ----------
        30       9400
        10       8750
```

Aggregate Functions

```
SQL> select DEPTNO , SUM(SAL) FROM EMP
        GROUP BY DEPTNO
          HAVING SUM(SAL) < 10000 ORDER BY DEPTNO;

    DEPTNO   SUM(SAL)
---------- ----------
        10       8750
        30       9400
```

```
SQL> select DEPTNO , SUM(SAL) FROM EMP
        GROUP BY DEPTNO
         HAVING SUM(SAL) < 10000
          ORDER BY SUM(SAL);

    DEPTNO   SUM(SAL)
---------- ----------
        10       8750
        30       9400
```

Group by

```
SQL> select deptno,count(*) from emp group by deptno;
```

DEPTNO	COUNT(*)
10	3
20	5
30	6

Conversion functions

YY

```
SQL> SELECT TO_CHAR(TO_DATE('23-DEC-25','DD-MON-YY'),'DD-MON-YYYY') FROM DUAL;

 DATE
23-DEC-2025

SQL> SELECT TO_CHAR(TO_DATE('23-DEC-49','DD-MON-YY'),'DD-MON-YYYY') FROM DUAL;
  DATE
23-DEC-2049

SQL> SELECT TO_CHAR(TO_DATE('23-DEC-50','DD-MON-YY'),'DD-MON-YYYY') FROM DUAL;

  DATE
23-DEC-2050

SQL> SELECT TO_CHAR(TO_DATE('23-DEC-77','DD-MON-YY'),'DD-MON-YYYY') FROM DUAL;

DATE
23-DEC-2077
```

```
SQL>  select SYSDATE FROM DUAL;
DATE
20-JUL-07

SQL> SELECT TO_CHAR(SYSDATE,'D') FROM DUAL;
DATE
6

SQL>  SELECT TO_CHAR(SYSDATE,'DD') FROM DUAL;
DATE
20

SQL>  SELECT TO_CHAR(SYSDATE,'DAY') FROM DUAL;
DATE
FRIDAY

SQL> SELECT TO_CHAR(SYSDATE,'Day') FROM DUAL;
DATE
Friday
```

```
SQL>  SELECT TO_CHAR(SYSDATE,'MM') FROM DUAL;
DATE
07

SQL> SELECT TO_CHAR(SYSDATE,'MON') FROM DUAL;
DATE
JUL

SQL>  SELECT TO_CHAR(SYSDATE,'Mon') FROM DUAL;

TO_
---
Jul

SQL> SELECT TO_CHAR(SYSDATE,'MONTH') FROM DUAL;

TO_CHAR(S
---------
JULY
```

```
SQL>  SELECT TO_CHAR(SYSDATE,'YY') FROM DUAL;
DATE
07

SQL> SELECT TO_CHAR(SYSDATE,'YYYY') FROM DUAL;
DATE
2007

SQL>  SELECT TO_CHAR(SYSDATE,`RR') FROM DUAL;
DATE
07

SQL> SELECT TO_CHAR(SYSDATE,`RRRR') FROM DUAL;
DATE
2007

SQL>SELECT HIREDATE,
TO_CHAR(HIREDATE,'DDD,DD/MM/YYYY HH:MI:SS') FROM EMP;

17-DEC-80 352,17/12/1980 12:00:00
20-FEB-81 051,20/02/1981 12:00:00
```

```
 SQL> select to_char(sysdate,'HH : MI : SS') from dual;

TO_CHAR(SYSD
------------
04 : 13 : 53

SQL> select to_char(sysdate,'HH12 : MI : SS') from dual;

TO_CHAR(SYSD
------------
04 : 14 : 08

SQL>  select to_char(sysdate,'HH24 : MI : SS') from dual;

TO_CHAR(SYSD
------------
16 : 14 : 19

SQL> select to_char(sysdate,'HH12 : MI : SS AM ') from dual;

TO_CHAR(SYSDATE,
----------------
04 : 15 : 06 PM

SQL> select to_char(sysdate,'HH24 : MI : SS AM ') from dual;

TO_CHAR(SYSDATE,
----------------
16 : 15 : 16 PM
```

To_Date(String, Str_Format)

```
SQL> SELECT * FROM EMP ORDER BY HIREDATE;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK          7902 17-DEC-80        800                    20
      7499 ALLEN      SALESMAN       7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN       7698 22-FEB-81       1250        500         30
      7566 JONES      MANAGER        7839 02-APR-81       2975                    20
      7698 BLAKE      MANAGER        7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER        7839 09-JUN-81       2450                    10
      7844 TURNER     SALESMAN       7698 08-SEP-81       1500          0         30
      7654 MARTIN     SALESMAN       7698 28-SEP-81       1250       1400         30
      7839 KING       PRESIDENT           17-NOV-81       5000                    10
      7900 JAMES      CLERK          7698 03-DEC-81        950                    30
      7902 FORD       ANALYST        7566 03-DEC-81       3000                    20
      7934 MILLER     CLERK          7782 23-JAN-82       1300                    10
      7788 SCOTT      ANALYST        7566 19-APR-87       3000                    20
      7876 ADAMS      CLERK          7788 23-MAY-87       1100                    20

14 rows selected.

SQL> SELECT * FROM EMP
            WHERE HIREDATE = '23-MAY-87';

EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7876 ADAMS      CLERK          7788 23-MAY-87       1100                    20
```

```
SQL> SELECT * FROM EMP
            WHERE HIREDATE = '23-MAY-87';

EMPNO ENAME   JOB     MGR HIREDATE     SAL    COMM   DEPTNO
----------- ---------- --------- ---------- --------- -------
7876 ADAMS    CLERK  7788 23-MAY-87   1100             20

SQL> SELECT * FROM EMP
          WHERE HIREDATE = TO_DATE('23-MAY-87');

EMPNO ENAME   JOB     MGR HIREDATE     SAL    COMM   DEPTNO
----------- ---------- --------- ---------- --------- -------
7876 ADAMS    CLERK  7788 23-MAY-87   1100             20

SQL> SELECT * FROM EMP
WHERE HIREDATE = TO_DATE('05-23-87','MM-DD-RR');

EMPNO ENAME   JOB     MGR HIREDATE     SAL    COMM   DEPTNO
----------- ---------- --------- ---------- --------- -------
7876 ADAMS    CLERK  7788 23-MAY-87   1100             20

SQL> SELECT * FROM EMP
WHERE HIREDATE = TO_DATE('05-23-1987','MM-DD-YYYY');

EMPNO ENAME   JOB     MGR HIREDATE     SAL    COMM   DEPTNO
----------- ---------- --------- ---------- --------- -------
7876 ADAMS    CLERK  7788 23-MAY-87   1100             20
```

```
 SQL>  SELECT * FROM EMP
WHERE HIREDATE = TO_DATE('MAY-23-87','MON-DD-YY');

no rows selected


SQL> SELECT TO_CHAR( TO_DATE('MAY-23-87','MON-DD-YY'),'YYYY')
FROM DUAL;
 Date
2087

SQL> SELECT TO_CHAR( TO_DATE('MAY-23-87','MON-DD-RR'),'YYYY')
FROM DUAL;
 Date
1987

SQL> SELECT * FROM EMP WHERE
HIREDATE = TO_DATE('MAY-23-87','MON-DD-RR');

EMPNO ENAME    JOB       MGR HIREDATE      SAL     COMM    DEPTNO
---------- ---------- --------- ---------- --------- -------
7876 ADAMS    CLERK   7788 23-MAY-87    1100              20
```

Set Operators

Commands :-

- UNION
- UNION ALL
- INTERSEC
- MINUS

Set Operators

UNION

- Combines the results of two select statements into one result set.
- Wont allow the Duplicates Records.

UNION ALL

- Combines the results of two select statements into one result set including the duplicates.

MINUS

Takes the result set of one SELECT statement, and removes those rows that are also returned by a second SELECT statement.

INTERSECT

- Returns only those rows that are returned by each of two SELECT statements.

Set Operators :-

- You can combine multiple Select queries using the set operators
- If a SQL statement contains multiple set operators, then Oracle Database evaluates them from the left to right unless parentheses explicitly specify another order.
- The number of columns and there Data type should be match in all the Queries.

Set Operators

UNION ALL

```
SQL> select * from dept
  2  UNION ALL
  3  select * from dept;

    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       DALLAS
        30 SALES          CHICAGO
        40 OPERATIONS     BOSTON
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       DALLAS
        30 SALES          CHICAGO
        40 OPERATIONS     BOSTON

8 rows selected.
```

Set Operators

UNION

```
SQL>  select * from dept
  2  UNION
  3  select * from dept;

    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       DALLAS
        30 SALES          CHICAGO
        40 OPERATIONS     BOSTON
```

Set Operators

MINUS

```
SQL>  SELECT EMPNO,ENAME,SAL,DEPTNO  FROM EMP
  2  MINUS
  3  SELECT EMPNO,ENAME,SAL,DEPTNO  FROM EMP WHERE DEPTNO=10;

    EMPNO ENAME           SAL     DEPTNO
--------- ---------- ---------- ----------
     7369 SMITH            800         20
     7902 FORD            3000         20
     7876 ADAMS           1100         20
     7566 JONES           2975         20
     7788 SCOTT           3000         20

     7844 TURNER          1500         30
     7900 JAMES            950         30
     7521 WARD            1250         30
     7499 ALLEN           1600         30
     7654 MARTIN          1250         30
     7698 BLAKE           2850         30

11 rows selected.
```

Set Operators

INTERSECT

```
SQL>  SELECT EMPNO,ENAME,SAL,DEPTNO  FROM EMP
  2  INTERSECT
  3  SELECT EMPNO,ENAME,SAL,DEPTNO  FROM EMP WHERE DEPTNO=10;

     EMPNO ENAME          SAL     DEPTNO
---------- ---------- ---------- ----------
      7782 CLARK          2450         10
      7839 KING           5000         10
      7934 MILLER         1300         10
```

Set Operators

@@

ERROR – Data Type

```
SQL>  select dname,deptno,loc from dept
  2     UNION
  3     select deptno,dname,loc from dept;
 select dname,deptno,loc from dept
        *
ERROR at line 1:
ORA-01790: expression must have same datatype as corresponding expression
```

@@

ERROR – Number of Columns

```
SQL> select dname,deptno,loc from dept
  2  UNION
  3   select dname,deptno FROM DEPT;
select dname,deptno,loc from dept
*
ERROR at line 1:
ORA-01789: query block has incorrect number of result columns
```

Set Operators

COLUMN NAME

```
SQL> select deptno as D ,dname,loc from dept
  2  UNION
  3  select * from dept;

         D DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       DALLAS
        30 SALES          CHICAGO
        40 OPERATIONS     BOSTON


SQL>  select * from dept
  2  union
  3  select deptno as D ,dname,loc from dept;

    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       DALLAS
```

Set Operators

Default Column

```
SQL>  select dname,deptno,loc from dept
  2   UNION
  3    select dname,deptno,NULL FROM DEPT;

DNAME          DEPTNO LOC
-------------- --------- ------------
ACCOUNTING             10 NEW YORK
ACCOUNTING             10
OPERATIONS             40 BOSTON
OPERATIONS             40
RESEARCH               20 DALLAS
RESEARCH               20
SALES                  30 CHICAGO
SALES                  30
```

Set Operators

ERROR - ORDER BY

```
SQL> select * from dept order by deptno
  2     UNION ALL
  3     select  * from dept order by deptno ;
ERROR at line 2:
ORA-00933: SQL command not properly ended
--------------------------------------------------------------------------------
   -
SQL> SELECT * FROM EMP
  2  INTERSECT
  3  SELECT * FROM EMP WHERE DEPTNO=10 ORDER BY DEPTNO;
ERROR at line 3:
ORA-00904: "DEPTNO": invalid identifier
--------------------------------------------------------------------------------
                                                    -
SQL>  SELECT DEPTNO,TO_NUMBER(2) AS NU FROM DEPT
  2  UNION
  3  SELECT DEPTNO,SUM(SAL) FROM EMP GROUP BY DEPTNO ORDER BY DEPTNO;
    DEPTNO         NU
---------- ----------
        10          2
        10       8750
        20          2
```

CONSTRAINTS

Types of constraints

PRIMARY KEY

- Will not allow duplicate values and NULL's are not allowed (NOT NULL & UNIQUE)
- Only one primary Kay can be defined in a table

UNIQUE

Will not allow duplicate values but allow NULL's .

NOT NULL

- Will not allow NULL values but allow Duplicates values.

FOREIGN KEY

- A column value which is derived from PRIMARY KEY column of same/another table.

CHECK

- Checks for the given condition before inserting the record into the Table

```
SQL> Alter Table Test ADD Constraint CPK Primary Key (eno);
SQL> Alter Table Test ADD Constraint CUK Unique  (ename);
SQL> Alter Table Test ADD Constraint CCK Check  (Sal   > 1000);
SQL> Alter Table Test ADD Constraint  CFK
                FOREIGN KEY (DEPTNO)   references DEPT(DEPTNO);

SQL> Alter Table Test  MODIFY  J_Date Varchar2(20) NOT NULL;
SQL> Alter Table Test  MODIFY  Deptno Number(8)    Default       10;

SQL> Alter Table Test Drop Primary Key;
SQL> Alter Table Test Drop Unique (ename);
SQL> Alter Table Test Drop  CONSTRAINT <C_NAME>;
SQL> Alter Table Test Drop  CONSTRAINT CPK ;


SQL> Alter Table Test ENABLE  Constraint <C_NAME>;
SQL> Alter Table Test  DISABLE Constraint <C_NAME>;
```

JOINS

JOINS

JOINS:

JOINs are used to retrieve information from multiple tables.

TYPES OF JOINS:

1. EQUI-JOIN :
2. NON-EQUI JOIN :
3. OUTER JOIN
    * LEFT OUTER JOIN
    * RIGHT OUTER JOIN
    * FULL OUTER JOIN
4. SELF JOIN

SELF JOIN

```
SQL> SELECT E.ENAME EMP_NAME,
  2  M.ENAME MGR_NAME
  3  FROM EMP E,EMP M
  4  WHERE E.MGR=M.EMPNO;

EMP_NAME   MGR_NAME
---------- ----------
SMITH      FORD
ALLEN      BLAKE
WARD       BLAKE
JONES      KING
MARTIN     BLAKE
BLAKE      KING
CLARK      KING
```

JOIN SYNTAX

Equi Join

```
SQL> SELECT Empno,Ename,Sal,Comm,Emp.Deptno,Dept.Deptno from
        Emp   JOIN, DEPT
        where EMP.deptno=DEPT.deptno;
```

Left Outer Join

```
SQL> SELECT Empno,Ename,Sal,Comm,Emp.Deptno,Dept.Deptno from
        Emp left OUTER JOIN DEPT
        ON EMP.deptno=DEPT.deptno;
```

Right Outer Join

```
SQL> SELECT Empno,Ename,Sal,Comm,Emp.Deptno,Dept.Deptno from
        Emp right OUTER JOIN DEPT
        ON EMP.deptno=DEPT.deptno;
```

Full Outer Join

```
SQL> SELECT Empno,Ename,Sal,Comm,Emp.Deptno,Dept.Deptno from
        Emp FULL OUTER JOIN DEPT
        ON EMP.deptno=DEPT.deptno;
```

SUB QUERY

Sub Query :-

- A query within a query called sub query.
- These subqueries are also called **Nested subqueries**.
- A subquery can also be found in the FROM clause. These are called **inline views**.

TYPES OF SUB QUERY :-

1. **Single Row** Sub Query
2. **Multi Row** Sub Query

Sub Query

- These subqueries can reside in the following Clauses
- **1) SELECT**
- **2) FROM -----(*In-line Query*)**
- **3) WHERE**
- **4) HAVING BY**
- **5) ORDER BY**
- **6) INSERT**
- **7) UPDATE**

- Most often, the subquery will be found in the WHERE clause.
- Oracle allows up to 255 levels of subqueries in the WHERE clause.
- Oracle allows an unlimited number of subqueries in the FROM clause.

Sub Query

SQL> select * from emp;

ENO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7788	SCOTT	ANALYST	7566	19-APR-87	3000		20
7839	KING	PRESIDENT		17-NOV-81	5000		10
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7876	ADAMS	CLERK	7788	23-MAY-87	1100		20
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7902	FORD	ANALYST	7566	03-DEC-81	3000		20
7934	MILLER	CLERK	7782	23-JAN-82	1300		10

Single Row Sub Query

- **SQL> select Min(sal) from emp;**
- MIN
- 800

- SQL> select * from emp where sal = **(select Min(sal) from emp);**
-
- ENO ENAME JOB MGR HIREDATE SAL COMM DEPTNO
- ---------- ---------- --------- ---------- --------- ---------- ---------- ----------
- 7369 SMITH CLERK 7902 17-DEC-80 800 20
- **(select Max(sal) from emp);**
- SQL> select * from emp where sal =
-
- ENO ENAME JOB MGR HIREDATE SAL COMM DEPTNO
- ---------- ---------- --------- ---------- --------- ---------- ---------- ----------
- 7839 KING PRESIDENT 17-NOV-81 5000 10
-
- SQL> select * from emp where sal <= **(select Avg(sal) from emp);**
-
- ENO ENAME JOB MGR HIREDATE SAL COMM DEPTNO
- ---------- ---------- --------- ---------- --------- ---------- ---------- ---------- ---------- ----------
- 7369 SMITH CLERK 7902 17-DEC-80 800 20
- 7499 ALLEN SALESMAN 7698 20-FEB-81 1600 300 30
- 7521 WARD SALESMAN 7698 22-FEB-81 1250 500 30
- 7654 MARTIN SALESMAN 7698 28-SEP-81 1250 1400 30
- 7844 TURNER SALESMAN 7698 08-SEP-81 1500 0 30
- 7876 ADAMS CLERK 7788 23-MAY-87 1100 20
- 7900 JAMES CLERK 7698 03-DEC-81 950 30
- 7934 MILLER CLERK 7782 23-JAN-82 1300 10

Multi Row Sub Query

SQL> select * from emp where deptno **in** (select deptno from dept);

```
ENO ENAME     JOB       MGR HIREDATE        SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ----------
      7369 SMITH      CLERK            7902 17-DEC-80       800                    20
      7499 ALLEN      SALESMAN         7698 20-FEB-81      1600        300         30
      7521 WARD       SALESMAN         7698 22-FEB-81      1250        500         30
      7566 JONES      MANAGER          7839 02-APR-81      2975                    20
      7654 MARTIN     SALESMAN         7698 28-SEP-81      1250       1400         30
      7698 BLAKE      MANAGER          7839 01-MAY-81      2850                    30
      7782 CLARK      MANAGER          7839 09-JUN-81      2450                    10
      7788 SCOTT      ANALYST          7566 19-APR-87      3000                    20
      7839 KING       PRESIDENT             17-NOV-81      5000                    10
      7844 TURNER     SALESMAN         7698 08-SEP-81      1500          0         30
      7876 ADAMS      CLERK            7788 23-MAY-87      1100                    20
      7900 JAMES      CLERK            7698 03-DEC-81       950                    30
      7902 FORD       ANALYST          7566 03-DEC-81      3000                    20
      7934 MILLER     CLERK            7782 23-JAN-82      1300                    10

14 rows selected.
```

SQL> select * from dept where deptno **in** (select Distinct(deptno) from emp);

```
    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       DALLAS
```

INDEX

INDEX:-

- An **index** is a performance-tuning method of allowing faster retrieval of records.
- An index creates an entry for each value that appears in the indexed columns.
- By default, Oracle creates B-tree indexes.
- You should have **INDEX** privilege on the table to be indexed.
- You should have **CREATE ANY INDEX** system privilege to create a Index.

The syntax for creating a index is:

SQL> Create [UNIQUE] index I_Name ON T_Name (C1 , C2 , C3)

INDEX

Types Of INDEX

1) UNIQUE INDEX
2) BEE TREE INDEX
3) BIT MAP INDEX
4) FUNCTIONED INDEX
5) REVERSE INDEX

VIEW

- A view is a logical table based on one or more tables.
- A view derives its data from the tables on which it is defined.
- A view does not hold any data.
- You can perform all DML (Insert , Update , Delete , Select) on View.
- All operations performed on a view actually affect the base table of the view.
- You can use views in almost the same way as tables.
- You can query, update, insert into, and delete from views, just as you can standard tables.

Note: Data can be added through a view unless it contains:

(Group by clause, rownum pseudo column, distinct keyword, Group function)

VIEW

```
SQL> Create View V1
        AS
           Select * from Emp ;
SQL> Create View V2
        AS
          Select * from Emp Where Deptno=10 ;
SQL> Create View V3
        AS
           Select Deptno , sum(sal) as SL from Emp Group By Deptno
SQL> Create View V4
        AS
          Select Eno ,Ename  from Emp ;
SQL> Create View V5
        AS
           Select Eno, Substr(Ename,1,3) AS NAME  from Emp ;
```

VIEW

View With Check Option ;-

```
SQL> Create View V1
        AS
            Select * from Emp
                    Where Deptno=10 WITH CHECK OPTION
```

- If you select the data from emp table you can see all the Dept values.
- If you select the data from V1 View you can see only Deptno =10 data.
- SPECIFIES THAT ONLY ROWS THAT WOULD BE RETRIVED BY THE QUERY CAN ONLY BE INSERTED , UPDATED OR DELETED.

VIEW

View With read only Option ;-

```
SQL> Create View V1 (Eno, sal )
        AS
            Select Deptno,Comm from Emp with READ ONLY ;
```

- You can only Read the Data.
- You can not Apply DML's on the VIEW.
- You can Apply DML's on the Base TABLE.

ORA-01733: virtual column not allowed here

WHY USE VIEWS AT ALL

VIEWS TO ENFORCE SECURITY:

IT MAY BE THAT USERS SHOULD ONLY SEE CERTAIN ROWS OR

COLUMNS OF A TABLE.

CONSIDER THE EMPLOYEE TABLE:

EMPLOYEE_ID
FIRST_NAME
LAST_NAME
EMAIL
PHONE_NUMBER
HIRE_ DATE
JOB_ID
SALARY
COMMISSION_PCT
MANAGER_ID
DEPARTMENT_ID

WHICH INCLUDES PERSONAL DETAILS THAT SHOULD NOT BE VISIBLE TO STAFF. BUT FINANCE STAFF WILL NEED TO BE ABLE TO SEE THE COSTING INFRMATION. BY USING VIEWS WE WILL DEPERSONALIZE THE DATA.

```
CREATE VIEW EMP_FIN
AS
SELECT HIRE_DATE,
        JOB_ID,
        SALARY,
        COMMOSSION_PCT,
        DEPARTMENT_ID
FROM EMPLOYEE
```

VIEWS TO SIMPLIFY USER SQL

IT WILL BE MUCH EASIER FOR USER TO QUERY DATA IF THE

HARD WORK (SUCH AS JOINS OR AGGREGATIONS) IS DONE FOR

THEM BY THE CODE THAT DEFINES THE VIEW.

```
SELECT
    E.Ename Employee,
    E.Sal "Employee's Salary",
    SE.Grade EmpGrade,
    M.Sal "Manager's Salary",
    SM.Grade MGRGrade,
    Dname
    FROM Emp E JOIN Dept D
    ON E.Deptno = D.Deptno  JOIN Emp M ON
    E.MGR = M.Empno JOIN SalGrade SE ON
    E.Sal BETWEEN SE.LoSal AND SE.HiSal JOIN SalGrade SM ON
    M.Sal BETWEEN SM.LoSal AND SM.HiSal
```

```
CREATE VIEW EMP_MGR_SALGRADE_INFO

AS

SELECT
    E.Ename Employee,
    E.Sal "Employee's Salary",
    SE.Grade EmpGrade,
    M.Sal "Manager's Salary",
    SM.Grade MGRGrade,
    Dname
    FROM Emp E JOIN Dept D
    ON E.Deptno = D.Deptno  JOIN Emp M ON
    E.MGR = M.Empno JOIN SalGrade SE ON
    E.Sal BETWEEN SE.LoSal AND SE.HiSal JOIN SalGrade SM ON
    M.Sal BETWEEN SM.LoSal AND SM.HiSal
```

TYPES OF VIEWS:

SIMPLE VIEWS:

A SIMPLE VIEW RETRIVES DATA FROM ONE DETAIL TABLE , USES NO FUNCTIONS, AND DOES NO AGGREGATIONS.

COMPLEX VIEW:

A COMPLEX VIEW CAN JOIN DETAIL TABLES,

USE FUNCTIONS, AND PERFORM AGGREGATIONS.

Materialized Views

Materialized Views ;-

- It is Static View, means it will hold the data.
- Used for Analyzing and Reporting.
- We need to have **Create Materialized View** Privilege for creating of it.

SQL FUNCTIONS:

****

SQL functions can be used to ..

- perform calculations on data
- modify individuals on data
- manipulate output for groups of rows
- format dates and numbers for display.
- convert column data type.

SQL functions types:

****

- single row functions
- multiple row functions

single row functions:

****

These functions return a single result for every row of a queried table.

multiple row functions:

****

These functions manipulate groups of rows and return one result per Group of rows.

Single row functions can appear in

SELECT List.

WHERE clause and order by clause.

Categories of single row functions:

****

Character functions

Number functions

Date functions

Conversion functions

Single row functions are used to manipulate data items.

Syntax:

****

FunctionName(column/expression, [arg1, arg2, ..])

Single row functions can be nested.

Specification behavior of functions:

****

Character functions:

accepts character input and return both character and number values.

the functions are categorized as:

1. case conversion functions
2. character manipulation functions

1. case conversion functions:

****

Lower function

Syntax: lower(column/expression)

Upper function

Syntax: Upper (column/expression)

Initcap function

Syntax:initcap (column/expression)

Concat Function

Concat(column1/expression1, column2/expression2)