25/07/2023
Tuesday

SQL
Structured Query Language

- SQL is also known as SEQUEAL.
- Data:
  - Data is a raw part which describes the properties ~~often~~ of an objects.
  - Properties are also known as attributes.
  - ~~entit~~ Objects is also known as entity.

Sandip
Entity/Object

Properties
- Name: Sandip Mandal
- Roll No - 24400319008
- D.O.B - 21.12.2001
- Father's Name - Biswajit Mandal
- Gen - Male
- Email Id - Sandipmandal4056@gmail.com
- Phone No - 6295510306
- AGE - 22 yrs.

Data

- Database:
  - CRUD operation
    - C - create / Insert
    - R - Read / Retrieve
    - U - update / Modify
    - D - Delete / Drop

> Database is a place or media in which we stores the data in a systematic an organised manner.

Data
+ Systematic
+ Organised

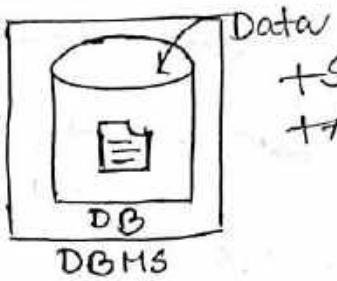
- In database we are gone a perform an operation which is known as "crud" operation.
  - C - Create / Insert
  - R - Read / Retrieve
  - U - Update / Modify
  - D - Delete / Drop.

realme Shot by ❤SANDIP❤

DBMS (Database Management System):

- DBMS is a software which is used to maintain and manage the database.
- Security and Authorization are the two main features provided by DBMS software.
- In DBMS software ~~which~~ we stores the data in from of "File" format.
- We used query language to communicate with DBMS software.

⇒ File
⇒ Query Language



+ Security
+ Authorization

Relational Database Management System (RDBMS)

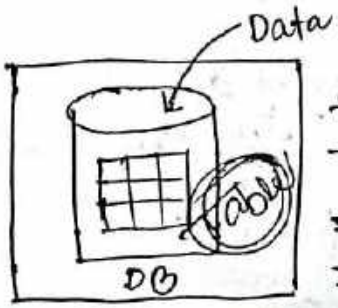
- RDBMS software which is used to maintain and manage the database

> • RDBMS software it is a type of DBMS software.

> • Security and Authorization are two main features provided by the RDBMS software.

> • In RDBMS software we stores the data in the form of "table" format.

> • We used SQL (Structured Query Language) to communicate with RDBMS software.



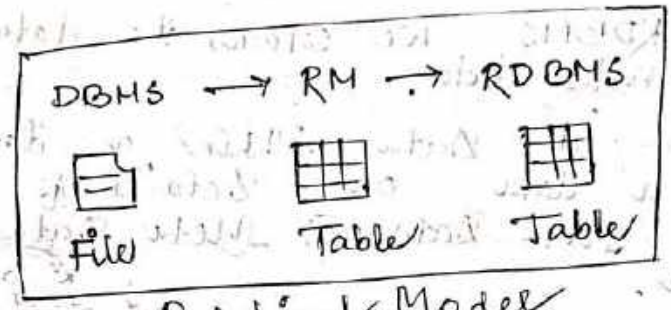
+ Security
+ Authorization

⇒ Table
⇒ SQL (Structured Query Language)

realme Shot by ❤SANDIP❤
realme 8i 2024 02 11 12:14

Relational Model (RM):

- Relational model is a theory proposed by ~~E.F Codd~~ one of the data Scientist known as E.F Codd
  - > In RM we stores the data in a "table" format
  - > Any DBMS which follows the relational model will be converted into RDBMS.
  - >



Relational Model

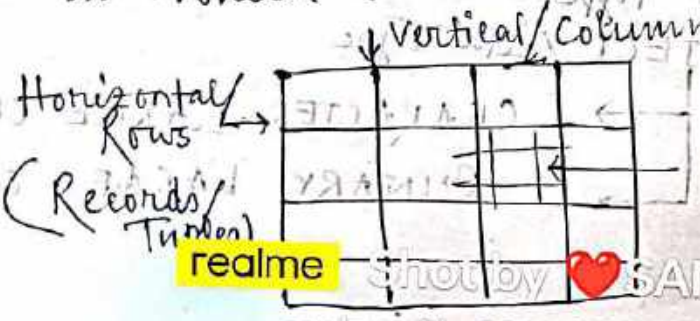
Table:

The combination of rows, columns and as well as cells is known as Table.

> **Rows:** ⊙ The Horizontal portion of table is known as rows.
⊙ Rows are also known as records/tuples.

> **Columns:** ⊙ The vertical portion of table is known as Columns.
⊙ Columns are also known as Attributes / Feilds.

> **Cells:** ⊙ The intersection of Rows and Columns is known as cell.
⊙ It is the smallest unit of a table, in which we stores the data.



realme Shot by ❤SANDIP❤
realme 8i 2024.02.11 12:14

Rules of E.F Codd:

① The data enter into the cells must be a single value.

② According to E.F Codd, we can store the data in multiple tables if we needed we can established the connection betn two tables with the help of Key attributes.

③ In RDBMS we stores the data in tables it includes meta data.

Meta Data: ① Data within a data or Data inside a data or Details of a data is known as Meta Data. ② Meta Data is auto generated.

IMG ID: IMG 001 Size: 4X4
Date: 26/07/23 Pixel: 1000
Time: 12:00 pm Location: JSP, BTR
Format: JPG

Meta Data

For example: images, meta data stores the size, encoding data, date of modification etc. The same is true for videos.

④ The data can made validated in two types:

i) By assinging data types
ii) By assinging ~~Constrients~~ Constraints constraints

Data types are mandatory, ~~Constrengths~~ are optional.

STU:

Photo	SID	SName	Branch	Phone No	Alt Ph no
	01	Sandip	ECE	999	
	02	Ramchandra	ECE	888	
	03	Manik	CSE	777	181
	04	Shubhadip	IT	666	

Data Types: Data Types are used to store what kind or what type of data can be stored in a particular memory location.

5 types of Data Types

① CHAR Data Type
② VARCHAR/ VARCHAR2 Data Type
③ NUMBER Data Type
④ DATE Data Type
⑤ LARGE OBJECT Data Type
→ CHARACTER LARGE Object
→ BINARY LARGE Object

realme Shot by SANDIP

CHAR Data Type:

> CHAR data types can accept 'A-Z', 'a-z', '0-9' 'Alphanumeric' & 'Special character'.

> Characters must have to be end close within Single Quote (' ').

> whenever we used CHAR datatype we have to mentioned the size.

① Syntax: CHAR (SIZE)

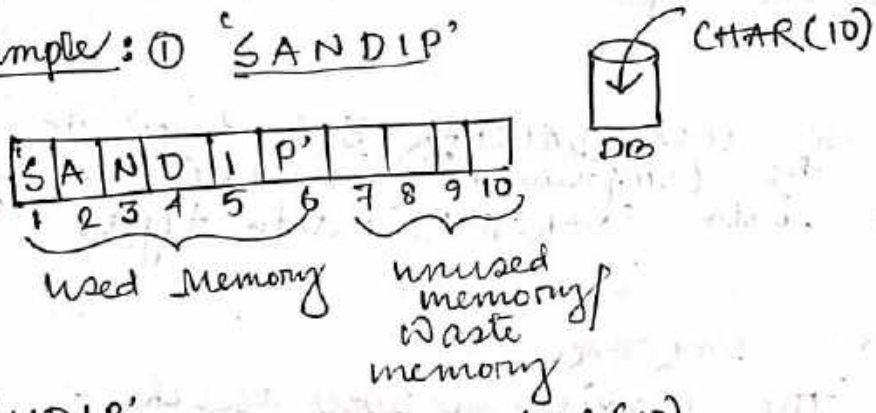
the max 2000ch

Fixed length memory allocation

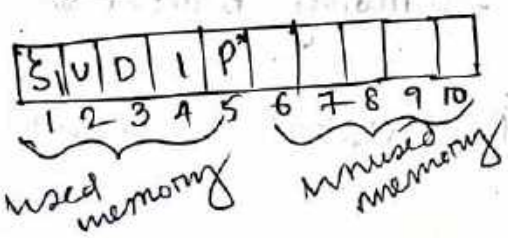
> the maximum size of CHAR data type is 2000ch.

> CHAR data types follows fixed length memory allocation.

Example: ① 'SANDIP'



② 'SUDIP'



VARCHAR Data Type:

> VARCHAR data types can accept 'A-Z', 'a-z', '0-9', 'Alphanumeric & Special character'.

> Characters must be have to be end close within Single Quote (' ').

> wh[illegible] VARCHAR data type we have to mentioned the size

realme Shot by realme 8i 2024.02.11 12:15

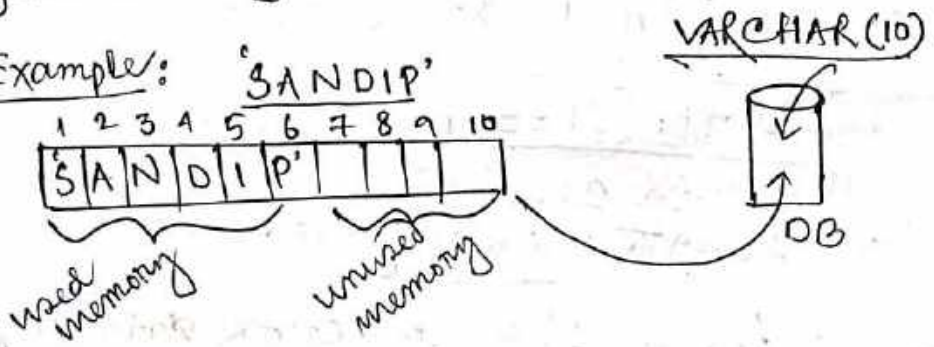
☑ Syntax: VARCHAR(SIZE)

The max 2000ch
Variable Length Memory Allocation

> The maximum size of VARCHAR data type is 2000ch

> VARCHAR data types follows the variable length memory allocation.

☑ Example: 'SANDIP'



☑ VARCHAR2 Data types

> It is the updated version of VARCHAR data type.

> If we pass VARCHAR data type to the column the compiler will automatically converts into 'VARCHAR2' data type.

(i) Syntax: VARCHAR(SIZE)

The maximum size 4000ch
Variable length memory Allocation

Q) Write the diffn betn CHAR, VARCHAR and VARCHAR2 data types.

CHAR	VARCHAR	VARCHAR2
① The maximum size of CHAR data type is 2000ch	① The maximum size of VARCHAR data type is 2000ch	① The maximum size of VARCHAR2 data type is 4000ch
② CHAR data types follows the fixed length memory allocation.	② VARCHAR data types follows the variable length memory allocation	② VARCHAR2 data types follows the variable length memory allocation.

realme
Shot by SANDIP

CHAR	VARCHAR	VARCHAR-2
③ Syntax: CHAR(SIZE)	VARCHAR(SIZE)	VARCHAR2(SIZE)

Number Data types

> It is used to store the numerical values.

Syntax: NUMBER (PRECISION [, SCALE])

[] ⇒ optional

Precision: i) It is used to store the integer values.
ii) The range of precision is 1 to 38.

Scale: i) It is used to store the decimal values.
ii) The range of scale is 0 to 127.
iii) The default value of scale is 0.

Example:

N(5,3) — P>S — 1 1 . 1 1 1

98754.235

P = 8 and S = 3 N(8,3)

9 8 7 5 4 2 3 5

N(3,5) — P<S — 0 0 1 1 1

N(3,3) — P = S — 1 1 1

N(5) — P = 5 — 1 1 1 1 1

Date Data Type

1. It is used to store the date formats.
2. It will be stored in a particular format known as "Oracle specified format".

realme Shot by SANDIP

Syntax: DATE

Format:

① 'DD-MON-YY'

(or)

② 'DD-MON-YYYY'

Example:

① '31-JUL-23'

② '31-JUL-2023'

Large Object Data Type

1. CHARACTER Large Object:

> It is used to store the large amount of characters upto 1GB of size.

2. BINARY Large Object:

> It is used to store the large amount of binary values upto 4GB of size.

Example: Images, mp3, mp4, documents, jpg, etc.

Constraints:

① Constraints are the rules given to the column for validations.

② Constraints are 5 types:

- → UNIQUE Constraints
- → NOT NULL Constraints
- → CHECK Constraints
- → PRIMARY KEY Constraints
- → FOREIGN KEY Constraints

UNIQUE:

Unique is a constraints which can't accept duplicate/repeated values.

NOT NULL:

Not Null is a constraints which can not accept null values.

realme Shot by SANDIP

☑ NULL: Null means empty values.

☑ NOTE: Zero(0) is not a null value. It is a value.

☑ CHECK:

① It is the extra ~~evalu~~ validation assigned to the column If the condition is satisfied It returns true.

> Syntax: CHECK (LENGTH (ColumnName) = SIZE)

> Example: CHECK (LENGTH (PhoneNo) = 10)

☑ PRIMARY KEY: It is the constraints which is used to identify a record, uniquely from a table.

⊙ Characteristics of Primary Key:

i) Primary Key can't accept duplicated/repeated values.

ii) Primary Key can't accept null values.

iii) It is the combination of "unique and not null" constraints.

iv) In the enter table we can assign only one column as a primary key.

v) Primary Keys not mandatory but highly recomended.

☑ FOREIGN KEY: It is a constraints which is used to established the connection between two tables.

⊙ Characteristics of Foreign Keys:

i) Foreign Key can accept duplicated/repeated values.

ii) Foreign Key can accept null values.

iii) It is not a combination of unique and not null constraints.

iv) [illegible] path to become a foreign key it is mandatory to be a

realme Shot by SANDIP realme 6i 2024 02 11 12:15

Primary Key it needs own table.

v) Either table became assign more than one Column as a foreign Key.

vi) Foreign Key is Present in child table but it belongs to parents table.

vii) Foreign Key is also known as referential constraints / referential integrity Constraints.

viii) It is not mandatory but highly recomended.

Emp : Primary Key

NN
UNQ
NN

CHECK
NN
UN

Foreign Key

EID	E Name	SAL	PHNO	DNO	SID
VARCHAR(4)	VARCHAR2(20)	Number(7)	Number(10)	N(5)	VARCHAR(10)
JSP01	SUN	60,000		10	1
JSP02	MOON	50,000		30	2
JSP03	STAR	40,000		10	3
JSP04	SKLAXY	30,000		20	4
JSP05	EARTH	20,000			5

Foreign Key

Dept : Primary Key

D.NO	D.Name	Location
10	D1	L1
20	D2	L2
30	D3	L3
40	D4	L4

STY Primary Key
SID S Name

SID	SName	Reg No	Branch
1	Sandip	101	ECE
2	Ram	102	ECE
3	Manik	103	CSE
4	Shubh	104	IT
5	Ritik	105	IT

Q.1) Diffn betn DBMS & RDBMS

Q.2) Diffn betn CHAR, VARCHAR and VARCHAR2 Data types

Q.3) Diffn betn Primary Key and Foreign Key

Q.4) Diffn betn Primary Key and UNIQUE Key Cons-
- traints.

Q.5) Diffn betn Foreign Key and UNIQUE Key Constraint

Customer Table:

Customer ID (Primary Key)	Customer Name	Phone No	Product ID (Foreign Key)
VARCHAR (10)	VARCHAR (20)	Number (10)	VARCHAR (10)
CUS101	SANDIP	6295503 06	PR01
CUS102	RAM	9784532 10	PR02
CUS103	MANIK	6295503 09	PR03
CUS103	RITIK	6951038 950	~~PR~~

Product Table:

Product ID (Primary Key)	Product Name	Location
VARCHAR (10)	VARCHAR (20)	VARCHAR (30)
PR01	Realme8i	G-PLOT
PR02	Redmi9i	East Medinipur
PR03	Redmi	West Medinipur

- Primary Key - CustomerID
- Foreign Key - ProductID

Q.1) Diff betn DBMS and RDBMS:

NO	DBMS	RDBMS
1.	DBMS stands for "Database Managment System".	1. RDBMS stands for "Relational Database Management System".
2.	DBMS technology stores the data in the form of file.	2. RDBMS stores the data in the form of tables.
3.	DBMS is designed to handle small amount of Data.	3. RDBMS is designed to deal with vast amount of data.
4.	DBMS provides support only for a single user at a time.	4. RDBMS provides supports for multiple users ~~for~~ at a time.
5.	Security is less.	5. More Security measures provided.
6.	We used query language to communicate with the DBMS software.	6. We used SQL (Structured Query Language) to communicate with the RDBMS' software.
7.	Examples: XML, Window Registry etc.	7. Examples: MySQL, PostgreSQL, SQL Server, Oracle etc.

realme Shot by SANDIP

Diffn betn CHAR, VARCHAR and VARCHAR2 Data types:

Sl.No	Purpose	CHAR	VARCHAR	VARCHAR2
1.	Storage Capacity	1. upto 2000ch	2. upto 2000ch	1. upto 4000ch
2.	Space	1. Follows the Fixed length Memory allocation	2. Follows the variable length memory allocation.	2. Follows the Variable length Memory allocation.
3.		3. It will waste space on the DISC memory.	3. This type of the data type is used to reserve the space for future usage.	3. The length of the variable strings values are stores within the values itself on the disc.
4.	Standard	4.	4. ANSI Standard	4. Oracle Standard
5.	Compare	5. Faster	5. Slower	5.
6.	Defaultsize	6. 1.	6. NO	6. NO

Diffn betn Primary Key and Foreign Key:

Sl.No	Primary Key	Foreign Key
1.	1. Primary Key can't accept duplicate/repeated values. It must contain unique values.	1. Foreign Key can accept duplicate/repeated values.
2.	Primary Key can't accept null values.	2. Foreign Key can accept null values.
3.	It is the combination of "UNIQUE and NOT NULL" constraints.	3. It is not a combination of "UNIQUE and NOT NULL" constraints.
4.	In the enter table we can assign only one column as a Primary Key.	4. A table became assign more than one column as a foreign Key.
5.	Primary Key is used to identify a record uniquely from a table	5. Foreign Key is used to established the connection between two tables.

realme Shot by

6.	It cannot create a parent-child relationship in a table.	6. It can make a parent-child relationship in a table.

Diffn betn Primary Key and Unique Key:

Sl No.	Primary Key	Unique Key
1.	1. Primary Key is used as a unique identifier for each record in the table.	1. The unique key is also a unique identifier for records when the primary key is not present in the table.
2.	We can not store NULL values in the Primary Key column.	2. We can store NULL value in the unique key column, but only one NULL is allowed.
3.	The Primary Key, by default, creates clustered index.	3. The unique key, by default creates a non-clustered index.
4.	Each table supports only one primary key.	4. A table can have more than one unique key.
5.	We cannot change or delete the primary key values.	5. We can modify the unique key column values.
6.	It is used to identify each record in the table.	6. It prevents storing duplicate entries in a column except for a NULL value.

Diffn betn Foreign Key and unique Key:

Sl. No	Foreign Key	unique Key
1.	1. Foreign key can accept multiple value	1. unique key may have a NULL value

realme Shot by SANDIP
realme 8i 2024 02 11 12:15

2. Foreign Key is a field in the table that is the Primary Key in another table.

2. Unique ~~Key~~ Constraint can not be related with another table's as a Foreign key.

3. We can have more than one foreign Key in a table.

3. Each table can have more than one unique Constraint.

4. Foreign Key do not automatically create an index, clustered or non-clustered. You can manually create an index on foreign Key.

4. By default, unique Key is a unique non-clustered index.

① Authors Table

NN UN Primary Key	UN	NN
AuthorId VARCHAR(10)	Name VARCHAR(30)	Country VARCHAR(20)
AU101	Howart	London
AU102	Stern	NewYork
AU103	Norton	America

Books Table

UN/NN	UN	Primary Key
BooksId VARCHAR(10)	BooksName VARCHAR(20)	AuthorId VARCHAR(10)
DS01	The Magic Tree	AU101
DS02	Winter Fairy	AU102
DBMS03	To work to the stars	AU103

Foreign Key

② Contacts Table

Primary Key UN/NN	UN	UN
ContactID VARCHAR(20)	EmailID VARCHAR(30)	FirstName
1	sandip@gmail.com	Sandip
2	Manik@gmail.com	Manik
3	Ram@gmail.com	Ram

Company Table

UN/NN	UN	Primary Key
CompanyID VARCHAR(10)	Company Name VARCHAR(30)	ContactID VARCHAR (10)
AB101	Awesome Company	1
AB102	OKay Company	2
AB103	OMG Company	3

Foreign Key

realme ❤️ SANDIP ❤️
realme 8i 2024 02 11 12:16

③ Country Table

Primary Key	NN	NN
CountryID VARCHAR(10)	Country VARCHAR(20)	Population Number(10)
1	India	1000000000
2	Australia	67538100000
3	Belgium	3257000000

City Table

Foreign Key

UN/NN	NN	Primary Key
cityID VARCHAR(20)	city VARCHAR(20)	CountryID VARCHAR(10)
11	Bengalore	1
13	Sydney	2
15	Ghent	3

④ Actor Table

UN/NN, Primary Key	UN	Foreign Key UN/NN
ActorID VARCHAR(10)	ActorName VARCHAR(20)	FilmID VARCHAR(10)
101	RAM	F101
102	MAHESH BABU	F102
103	RAM CHARAN	F103

Film Table

Primary Key UN		UN/NN
FilmID VARCHAR(10)	FilmName VARCHAR(20)	Title VARCHAR(30)
F101	Smart Sanker	The Deported
F102	Secret Spy	True Grit
F103	Betting Raja	No Time to die

⑤ Cricket-I Table

UN/NN Primary Key	UN	Foreign Key UN/NN
PlayerID VARCHAR(10)	Player Name VARCHAR(20)	JerseyNo Number(10)
PL01	MS Dhoni	7
PL02	Virat Kholi	18
PL03	Rabindra Jadeja	8

Cricket-II Table

UN/NN Primary Key	NN	UN/NN
JerseyNo Number(10)	Runs Number(7)	Popularity Number(10)
7	183	55
18	157	23
8	96	12

realme Shot by ❤SANDIP❤
realme 8i 2024 02 11 12:16

03/23 Thursday

Overview of SQL Statements:

1. DDL (Data Definition Language)
2. DML (Data Manipulation Language)
3. DCL (Data Control Language)
4. TCL (Transaction Control Language)
5. DQL / DRL (Data Query Language / Data Retrieve Language)

DQL (Data Query Language):

1. It is used to retrieve the data from the database.
2. It has 4 statements:
   a) SELECT
   b) PROJECTION
   c) SELECTION
   d) JOINS

a) SELECT: It is the process of retrieve / fetch the data from the table and display.

b) PROJECTION: It is the process of retrieve / fetch the data from the table by selecting only columns is known as Projection.

c) SELECTION: It is the process of retrieve / fetch the data from the table by selecting both rows as well as columns is known as Selection.

d) JOINS: It is the process of retrieve / fetch the data from multiple tables simultaneously is known as Joins.

realme Shot by SANDIP

Projection:

It is the process of retrieve / fetch the data from the table by selecting only columns is known as Projection.

- ~~Syntax~~:

Write a Query to displays names of employees.

Syntax:

```
SELECT * / [DISTINCT] Col-name / Expression [ALIAS]
from Table-Name;
```

EMP o/p of from clause

EMPNO	EName	SAL	DEPNO
1	A	100	10
2	B	200	30
3	C	300	20
4	D	400	10
5	E	500	30

under execution

- Order of execution

From

Select

⇒ Select EName from EMP;

- Result:

EName
A
B
C
D
E

o/p of Select clause

- Note: 1. From ~~class~~ clause starts the execution. In from clause we are passing table name as an arguments.

2. The work of from clause is go to the database search for the table and put the table under execution.

3. Select clause will executes after the execution of from clause

4. In the select clause we pass three arguments. the three arguments are:

a) *
b) Column-name
c) Expression [ALIAS]

realme ❤ SANDIP ❤
realme 8i · 2024 02 11 12:16

5) The job of Select Clause is go to the table which is under execution and select the columns which are mentioned.

6) Select Clause is response for the prepare in the result table.

Asterisk
~~Astricts~~ (*): It is mean to select all the columns from the table.

Semi-colon (;) : It is used to end the Query.

Set pages 100 lines 100 used to set the length and width of the page.

Examples:

① WAQTD Salary of employees.
→ Select SAL from EMP;

② WAQTD name and Salary of employees.
→ Select ENAME, sal FROM emp;

③ WAQTD Commision of employees
→ Select COMM from emp;

④ WAQTD employee no, employe name, job, mgr Hiredate, Sal, and Comm of employees.
→ ~~Select * from emp~~;
Select ename, empno, job, mgr, hiredate, Sal, Comm from emp;

⑤ WAQTD all the Details from the employee Table.
→ Select * from emp;

⑥ WAQTD names of all the employees.
→ Select ename from emp;

⑦ WAQTD Name and Salary Given to all the employee
→ Select ename, sal from emp;

⑧ WAQTD na[illegible] Commission Given to [illegible] the employ
→ Select ename, Comm from emp;

realme

(9) WAQTD Employee ID and department Number of all the employees

→ Select empno, ~~deptnum~~ deptno from emp;

(10) WAQTD Ename and Hiredate of all the employees.

→ Select ename, hiredate from emp;

(11) WAQTD Name and designation of all the employees

→ Select ename, ~~design~~ job from emp;

(12) WAQTD Name, Job and Salary given all the employees.

→ Select ename, job, sal from emp;

(13) WAQTD DNames present in the department table.

→ Select dname from dept;

(14) WAQTD dname and Location present in dept table.

→ Select dname, loc from dept;

Distinct: ① It is used to remove the duplicated/repeated values from the table.

② We have to pass distinct as a first argument in Select clause.

③

① WAQTD Salary of employees without repetition.

→ Select distinct sal from emp;

Q.1) WAQTD ename and Salary of employees without repeatition.

→ Select distinct ename, sal from emp;

ALIAS: ① It is used to change the column name in the result table.

② We can assign the alias name to the column with or without using As keyword.

realme Shot by SANDIP
realme 8i 2024 02 11 12:16

③ The ALIAS name has to be a single string and the string can be separated by using underscore (_) or by using double quote (" ").

Q.1) WAQTD ename of employees change the column name into employees.

→ Select ename as "employees" from emp;
/ Select ename employees from emp;

Q.2) WAQTD ename, sal, job, hiredate, department number, commission of employees change the ename column into Sharuk Khan, Salary column into Suresh, job column into fish, hiredate column into ~~mysho~~ change department number column into Samantha, and commission column into mango.

→ Select ename as "Sharuk Khan", sal as "Suresh" job as "fish", hiredate as "myshore", deptno as "Samanthia", comm as "mango" from emp;

~~Q.3) WAQTD~~

Expression: The statement which gives the results is known as expression.

Example: 2 + 3 = 5 → Result

operands

Statements

operators

In the expression we have a two arguments.

i) Operator
ii) operands

i) Operator: It is a arguments which performs the operation.

ii) Operands: ~~It is a~~ This are the arguments which involve the operations.

realme Shot by ❤SANDIP❤
realme 8i 2024 02 11 12:16

Q) WAQTD Salary of employees.
Select sal from emp;

Q) WAQTD Annual Salary of employees.
Select sal*12 from emp;

Q) WAQTD Half term / mid term Salary of employees.
Select sal* 6 from emp;

Q) WAQTD Quater Salary of employees.
Select sal* 3 from emp;

Q) WAQTD Salary of employees with increment of 100 Rs.
Select sal+100 from emp;

Q) WAQTD annual Salary of employees with increment of 1 Rs.
Select (sal*12)+1 from emp;

Q) WAQTD nine month Salary of employees with decrement of 5 Rs.
Select (Sal*9)-5 from emp;

Q) WAQTD details of employees with their annual Salary.
* ~~Select *, Sal*12 from emp;~~
Select emp.*, sal*12 from emp;

Q) WAQTD Salary of employees with increment of 10%.
Sal+Sal*10/100
~~Select emp.*~~ Select sal + sal* 10/100 from emp;
/ Select sal + sal* 0.1 from emp;

Q) WAQTD Salary of employees with decrement of 25%.
Select sal - sal* 25/100 from emp;

Q) WAQTD Annual Salary of employees with increment of 0.1% in the Annual Salary.
Select (sal + sal* 0.1/100)*12 from emp;

realme
Shot by SANDIP
realme 8i 2024 02 11 12:16

Q) WAQTD annual Salary of employees with decrement of 10%.

Select Sal*12 - Sal*12 * $\frac{10}{100}$ from emp;

Q) WAQTD midterm Salary of employees with increment of 7%.

Select Sal*6 + Sal*6 * $\frac{7}{100}$ from emp;

Q) WAQTD details of employees with their annual Salary increment of 500 Rs.

Select emp.*, Sal*12+500 from emp;

Q) WAQTD names of employees with their monthly Salary increment of 10% change the ename column into employeename.

Select ename as employeename, Sal + Sal* 10/100 from emp;

Q) WAQTD Salary of employees with their daily Salary.

Select Sal, Sal/30 from emp;

Q) WAQTD ename of employees with their details and increment of 10% in their 7 month Salary, decrement of 1% in their Ralf term Salary and increment of 100 Rs. in their daily Salary and decrement 1 Rs in their daily Salary and change the midterm Salary column name into Ralfterm, 7 month Salary Column name into Proun and ename Column name into light.

Select ~~ename~~ emp.*, Sal*7 + Sal*7 * $\frac{100}{100}$ as Proun, ~~Sal*6 - Sal*6 * 1/100~~, ~~Sal/30+100, Sal/30-1~~,

Sal* 6 - Sal* 6 * $\frac{1}{100}$ as Ralfterm, Sal/30 +100, Sal/30-1, ename as light from emp;

realme Shot by ❤SANDIP❤
realme 8i 2024 02 11 12:16

① WAQTD Name of employee Along with their annual Salary.

Select ename, sal * 12 from emp;

② WAQTD Ename and job for all the employees with their half term salary.

Select ename, job, sal * 6 from emp;

③ WAQTD ALL the details of the employees along with an annual bonus of 2000.

Select emp.*, sal * 12 + 2000 from emp;

④ WAQTD Name salary and salary with a hike of 10%

Select ename, sal, sal + sal * 10/100 from emp;

⑤ WAQTD Name and salary with deduction of 25%

Select ename, sal - sal * 25/100 from emp;

⑥ WAQTD Name and salary with monthly hike of 50.

Select ename, sal + 50 from emp;

⑦ WAQTD Name and annual salary with deduction of 10%

Select ename, sal * 12 - sal * 12 * 10/100 from emp;

⑧ * WAQTD Total salary given to each employee (SAL+COMM)

⑨ WAQTD Details of all the employees along with annual salary.

Select e[illegible] sal * 12 from emp;

realme Shot by ❤ SANDIP ❤

(10) WAQTD Name and designation along with 100 penalty in Salary.
Select ename, job, Sal-100 from emp;

Selection: It is the process of retrieving the data by selecting both rows and columns

(i) Syntax:

```
Select */[distinct] Col-name/Expression[Alias]
from Table-Name,
where <filter-Condition>;
```

(ii) Order of execution:

1. from class
2. where class
3. and Select Class

(1) WAQTD the name of the employee working in deptno 10.

→ ~~Select ename where deptno=10 from emp;~~
Select ename from emp where deptno=10;

(2) WAQTD name and salary of the employee working the deptno 30.

→ Select ename, Sal from emp where deptno=30;

(3) WAQTD the details of the employee who are working as salesman.

~~Select * emp, where ename=~~
→ Select * from emp where job='SALESMAN';

(4) WAQTD name of the employee who are working as clerk.

→ Select ename from emp where job='CLERK';

(5) WAQTD the name of employee who was hired on the date 20-Feb-1981.

→ Select ename from emp where HIREDATE='20-FEB-81';

⑥ WAQTD Details of employee who are getting salary greater than 3000.

→ Select * from emp where SAL > 3000;

⑦

(i) operators:

(ii) WHERE CLAUSE:

① WAQTD the annual salary of the employee whose name is Smith.

→ Select ~~ename~~ salary * 12 from emp where ename = 'SMITH';

④ WAQTD Details of employee who earns more than 2000.

→ Select * from emp where SALARY > 2000;

⑤ WAQTD details of employee whose name is Jones.

→ Select * from emp where ename = 'JONES';

⑥ WAQTD details of employee who has hired after 01-JAN-81.

→ Select * from emp where hiredate > '01-JAN-81';

⑦ WAQTD Name and salary along with his annual salary if the annual salary is more than 12000.

→ ~~Select emp.*, ename, salary, where salary * 12 > 12000;~~

→ Select ename, salary, salary * 12 from emp where salary * 12 > 12000;

⑧ WAQTD empNo of the employees who are working in Dept 30.

→ Select empno from emp where Deptno = 30;

⑨ WAQTD ename and hiredate if they are hired before 1981.

→ Select ename, hiredate from emp where hiredate < '01-JAN-1981';

⑩ WAQTD details of employee working as manager.

→ Select * from emp where Job = 'MANAGER';

realme Shot by SANDIP
realme 6i 2024.02.11 12:17

Operators:

① Arithmetic operator (+, -, *, /)
② Concatenation operator (||)
③ Comparision Operator (=, != (or) <>)
④ Relational Operator (>, <, >=, <=)
⑤ Logical operator (And, OR, NOT)
⑥ Special operator - i) IN ii) NOT IN iii) Between iv) Not between v) Is vi) Is Not vii) Like viii) Not Like
⑦ SubQuery operator - i) All ii) Any iii) Exists iv) Not Exists.

concatenation Operator(||): This operator is used to merch two or more strings

Example: Select 'MR.' || ename from emp;

o/p Table

'MR.' || ename
MR. SMITH
MR. ALLEN
MR. JONES
MR. CLERK

⑪ WAQTD name and salary given to an employee if employee earns a commission of rupees 1400.
→ Select ename, sal from emp where Comm = 1400;

⑫ WAQTD details of employees having commission more than salary.
Select * from emp where Comm > salary;

⑬ WAQTD empno of employees hired before the year 87.
Select empno from emp where hiredate < '01-JAN-87';

⑭ WAQTD details of employees working as an analyst
Select * from emp where job = 'ANALYST';

⑮ WAQTD details of emps earning more than 2000 rupees per month
Select * from emp where sal > 2000;

realme Shot by ❤ SANDIP ❤
realme 8i 2024 02 11 12:17

EMP Table:

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPT NO
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7499	ALLEN	SALEMAN	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	ADMAS	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

Logical Operator: ① Logical operators are used to write multiple filter condition.

② we have three type of logical operators -

- i) AND
- ii) OR
- iii) NOT.

i) **AND operator:** AND operator returns true if all the given conditions have satisfied.

① WAQTD details of employees who are working as clerk in department no 10.

→ Select * from emp where job = 'CLERK' and deptno = 10;

② WAQTD name, job, salary of employees who are working as clerk and getting salary less than 1000 Rs.

→ Select ename, job, sal from emp where job = 'CLERK' and SAL < 1000;

realme
Shot by SANDIP
realme 8i 2024 02 11 12:17

ii) OR operator: ① OR operator returns true if any one of the given condition has satisfied.

① WAQTD name, job, deptno of employees who are working in deptno 30 or working as Manager.

→ Select ename, job, deptno from emp where job = 'Manager' OR deptno = 30;

② WAQTD name, salary of employees who are working in deptno 40 and salary > 10Rs.

→ Select ename, sal from emp where deptno = 40 AND sal > 10;

③ WAQTD details of employees who are working in deptno 10, 20.

→ Select * from emp where deptno = 10 OR 20;

④ WAQTD name, job, salary, deptno of employees who are working in deptno = 10, 30 and working as salesman.

→ Select ename, job, sal, deptno from emp where (deptno = 10 OR 30) AND job = 'SALESMAN';

⑤ WAQTD details of employees increment of 1% in their salary and who are getting salary more than 1000 Rs. and working in deptno = 20.

Select emp.*, sal + sal * 1/100 where sal > 1000 AND deptno = 20;

realme Shot by ❤SANDIP❤
realme 8i 2024 02 11 12:17

iii) NOT operator: i) NOT operator is a unary operator instead of selecting the values, it will reject the value.

Q.1) WAQTD name, job of employees who are not working as manager and not working as Analyst and not working as President.

→ Select ename, job from emp
where job = 'MANAGER' NOT
~~job = 'ANALYST' NOT~~
~~job = 'PRESIDENT'~~ ;

Select ename, job from emp
where job <> 'MANAGER' AND job != 'ANALYST' AND not job = 'PRESIDENT';

Q.2) WAQTD details of employees who are working in deptno = 10 not working in deptno = 20 working as manager and not working as clerk and getting salary more than 1500 and hired after 1910.

→ Select * from emp where deptno = 10 AND NOT deptno = 20 AND job = 'MANAGER'
AND NOT job = 'CLERK' AND sal > 1500
AND hiredate > '31-DEC-1910';

Q.3) WAQTD name, salary of employees without repeatition who are working as manager, salesman, analyst and not working in deptno 30, 40.

→ Select distinct ename, sal from emp
where (job = 'MANAGER' ~~AND~~ OR
job = 'SALESMAN' ~~AND~~ OR
job = 'ANALYST') ~~AND~~ ~~OR~~ AND
~~job~~ NOT (deptno = 30 ~~AND~~ 40);

realme Shot by ❤SANDIP❤
realme 6i 2024 02 11 12:17

Logical operator Assignment

① WAQTD details of the employees working as clerk and earning less than 1500.

```
Select * from emp where job = 'CLERK' AND
    sal <1500;
```

② WAQTD name and hiredate of the employees working as manager in Dept 30.

```
Select ename, hiredate from emp where
job = 'MANAGER' AND deptno = 30;
```

③ WAQTD details of the employee along with annual salary if they are working in dept 30 as salesman and their annual salary has to be greater than 14000.

```
Select emp.*, sal*12 from emp where deptno = 30
AND job = 'SALESMAN' AND sal*12 > 14000;
```

④ WAQTD all the details of emp working in dept 30 or as analyst.

```
Select * from emp where deptno = 30 OR
    job = 'ANALYST';
```

⑤ WAQTD names of the employees whos salary is less than 1100 and their designation is clerk.

```
Select ename from emp where sal <1100
AND job = 'CLERK';
```

⑥ WAQTD name and sal, annual sal and deptno if deptno is 20 earning more than 1100 and annual salary exceeds 12000.

```
Select ename, sal, sal*12, deptno from emp
where deptno = 20 AND sal > 1100 AND
    sal*12 > 12000;
```

⑦ WAQTD empno and names of the employees working as manager in dept 20.

```
Select empno, ename from emp where deptno = 20
AND job = 'MANAGER';
```

⑧ WAQTD details of employees working in dept 20 or 30.

```
Select * from emp where deptno = 20 OR
    deptno = 30;
```

realme Shot by ❤SANDIP❤
realme 8i 2024 02 11 12:17

(9) WAQTD details of employees working as analyst in dept 10.

Select * from emp where job = 'ANALYST' AND deptno = 10;

(10) WAQTD details of employees working as president with salary of rupees 4000.

Select * from emp where job = 'PRESIDENT' AND sal = 4000;

(11) WAQTD names and deptno, job of emps working as clerk in dept 10 or 20.

Select ename, deptno, job from emp where job = 'CLERK' AND (deptno = 10 OR deptno = 20);

(12) WAQTD details of employees working as clerk or manager in dept 10.

Select * from emp where (job = 'CLERK' OR job = 'MANAGER') AND deptno = 10;

(13) WAQTD names of emps working in dept 10, 20, 30, 40.

Select ename from emp where deptno = 10 OR deptno = 20 OR deptno = 30 OR deptno = 40;

(14) WAQTD details of emps with empno 7902, 7839.

Select * from emp where empno = 7902 OR empno = 7839;

(15) WAQTD details of emps working as manager or salesman or clerk.

Select * from emp where job = 'MANAGER' OR job = 'SALESMAN' OR job = 'CLERK';

(16) WAQTD names of emps hired after 81 and before 87.

Select ename from emp where hiredate > '31-DEC-1981' AND hiredate < '01-JAN-1987';

(17) WAQTD details of emps earning more than 1250 but less than 3000.

Select * from emp where sal > 1250 AND sal < 3000;

(18) WAQTD names of emps hired after 81 into dept 10 or 30.

Select en[illegible] or emps [illegible] hiredate > '31-DEC-1981' AND (deptno = 10 OR deptno = 30);

realme
Shot by SANDIP
2024 02 11 12:17

(19) WAQTD names of employees along with annual salary for the emps working as manager or clerk into dept 10 or 30.

Select ename, sal*12 from emp where (job='MANAGER' OR job='CLERK') AND (deptno=10 OR deptno=30);

(20) WAQTD all the details along with annual salary if sal is between 1000 and 4000 annual sal more than 15000.

Select emp.*, sal*12 from emp where (sal>1000 AND sal<4000) AND sal*12>15000;

Special operator:

1. In
2. Not In
3. Between
4. Not Between
5. Is
6. Is Not
7. Like
8. Not Like

(1) **In operator**: In operator is a multivalued operator. It will select single value at LHS and multi values at RHS.

Syntax:

Column-Name/Expr IN (v1, v2, v3, ... vn);

(1) WAQTD details of employees who are working in deptno 10, 20, 30, 40.

→ Select * from emp where (deptno = 10 OR deptno = 20 OR deptno = 30 OR deptno = 40);

→ Select * from emp where deptno IN (10, 20, 30, 40);

(2) WAQTD name and job of emps who are working as clerk, manager, analyst.

Select ename, job from emp where job IN ('CLERK', 'MANAGER', 'ANALYST');

realme Shot by ❤SANDIP❤
realme 8i 2024 02 11 12:17

(I) **Not In operator**: Not In operator is similar to In operator instead of Selecting the values it will reject the values.

Syntax:

```
Column-name / Expr NOT IN (V1, V2, V3 ... Vn);
```

(1) WAQTD name and deptno of emps who are not working in deptno 10, 20, 40, 50.

→ Select ename, deptno from emp where deptno NOT IN (10, 20, 40, 50);

(2) WAQTD name and job of emps who are not working as Salesman, President.

→ Select ename, job from emp where job NOT IN ('SALESMAN', 'PRESIDENT');

(3)

(II) **Between operator**: i) Between operator is used whenever we have a range of values

ii) The ranges like High-range and low range or Stop range and Start range.

(II) Syntax:

```
Column-name / Expr BETWEEN 'Start-Range'
AND 'Stop-Range';
```

(1) WAQTD name and salary of employees who are getting salary in the range of 1000 and 3000.

→ Select ename, sal from emp where sal BETWEEN 1000 AND 3000;

(2) → Select ename, sal from emp where sal > 1000 AND sal < 3000;

(2) WAQTD name and Hiredate of emps who are hired in the range of 1980 to 1988.

Select Hiredate, from emp where Hiredate BETWEEN ~~1980~~ '01-JAN-1980' AND ~~1988~~ '31-Dec-1988';

realme Shot by SANDIP realme 8i 2024/03/31 12:17

③ WAQTD details of emps who are hired in the year of 1987

→ Select * from emp where hiredate ~~'01-JAN-1987'~~;
Between '01-JAN-1987' AND '31-Dec-1987';

④ WAQTD name and salary of emps who are getting salary in the range betn 1000 and 3000.

→ Select ename, sal from emp where
sal Between 1000 and 2999;

⑤ WAQTD name and hiredate of emps who are hired in the range betn 1981 to 1987.

→ Select ename, hiredate from emp where
hiredate between '01-JAN-1982' AND
'31-Dec-1987';

NOT Between Operator: NOT Between operator is similar to Between operator instead of selecting the values it will reject the values.

Syntax:

```
Column-name/Expr NOT BETWEEN
    'Start-Range' AND 'Stop-Range';
```

① WAQTD details of emps who are not getting salary in the range of 1000 to 4000.

→ Select * from emp where sal
NOT Between 1000 AND 4000;

② WAQTD hiredate of emps who are not hired in the range betn 1950 to 2050

→ Select hiredate from emp where
hiredate NOT BETWEEN '01-JAN-1951'
AND '31-Dec-2049';

realme
Shot by SANDIP
realme 8i 2024.02.11 12:18

Is operator: Is operator is used to compare with null values.

Syntax:

Column-name/Expr IS Null;

① WAQTD name, sal, and comm of emps who are not getting Commission.

→ Select ename, sal, comm from emp where comm is null;

② WAQTD name and mgr of employees who are not having reporting manager.

→ Select ename, mgr from emp where mgr is null;

IS NOT operator: Is not operator is similar to Is operator instead of selecting the values it will reject the values.

Syntax:

Column-name/Expr ISNOT Null;

① WAQTD names and comm of emps who are getting comm.

→ Select ename, comm from emp where comm ISNOT Null;

② WAQTD name and mgr of emps who are having reporting manager.

→ Select ename, mgr from emp where mgr ISNOT null;

LIKE operator: i) LIKE operator is used whenever we have pattern matching.

ii) To open the pattern matching two special characters - ⓐ (%) Persentile ⑤ (_) (underscore)

realme Shot by ❤ SANDIP ❤
realme 8i 2024 02 11 12:18

Syntax:

Column-name/Expr LIKE 'Pattern-To-Match';

① WAQTD name of emps who's name starts with A.

→ Select ename from emp where ename LIKE 'A%';

② WAQTD name of emps whose name ends with S.

→ Select ename from emp where ename LIKE '%S';

③ WAQTD names of emps whose name has character N.

Select ename from emp where ename LIKE '%N%';

④ WAQTD names of emps whose name has exactly 4 characters.

Select ename from emp where ename LIKE '----';

⑤ WAQTD name of emps whose name has last ~~two~~ 2nd character is E.

Select ename from emp where ename LIKE '%E-';

⑥ WAQTD names of emps whose name has last bit 2nd character is L.

Select ename from emp where ename LIKE '%L--';

realme Shot by ❤SANDIP❤
realme 8i 2024 02 11 12:18

Special operator

① List all the emps whose commission is null.
Select ename from emp where comm is null;

② List all the emps who don't have a reporting manager.
Select ename from emp where mgr is null;

③ List all the Salesman in dept 30
Select ename from emp where job='SALESMAN' AND deptno IN 30;

④ List all the Salesman in dept number 30 and having Salary greater than 1500.
Select ename from emp where job='SALESMAN' AND deptno=30 AND sal>1500;

⑤ List all the employees whose name starts with 'S' OR 'A'.
Select ename from emp where ename LIKE 'S%' OR ename LIKE 'A%';

⑥ List all the emps except those who are working in dept 10 & 20.
Select ename from emp where deptno not in (10,20);

⑦ List the emps whose name does not start with 'S'.
Select ename from emp where ename not like 'S%';

⑧ List all the emps who are having reporting managers in dept 10.
Select ename from emp where mgr is not null AND deptno=10;

⑨ List all the emps whose comm is null and working as clerk.
Select ename from emp where comm is null and job='CLERK';

⑩ List all the emps who don't have a reporting manager in deptno 10 or 30.
Select ename from emp where mgr is null and (deptno=10 OR deptno=30);

realme Shot by ❤SANDIP❤
realme 8i 2024 02 11 12:18

(11) List all the salesman in dept 30 with sal more than 2450.

Select ename from emp where job = 'SALESMAN' AND deptno = 30 AND sal > 2450;

(12) List all the analyst in dept number 20 and having salary greater than 2500.

Select ename from emp where job = 'ANALYST' AND deptno = 20 And sal > 2500;

(13) List all the emps whose names starts with 'M' OR 'J'.

Select ename from emp where ename LIKE 'M%' OR LIKE 'J%';

(14) List all the emps with annual salary except those who are working in dept 30.

Select ename, sal* 12 from emp where deptno not in 30;

(15) List all the emps whose name does not end with 'ES' OR 'R'

Select ename from emp where ename not like '%ES' OR not like '%R';

(16) List all the emps who are having reporting managers in dept 10 along with 10% hike in salary.

Select ename, sal + sal*0.1 from emp where mgr is not null and deptno=10;

(17) * Display all the employees who are salesman having 'E' as the last but one character in ename but salary having exactly 4 character.

Select ename from emp where job = 'SALESMAN' AND ename like '%E_' AND SAL LIKE '----';

(18) * Display all the emps who are joined after year 81.

Select ename from emp where Hiredate > '31-Dec-81';

(19) * Display all the emps who are joined in feb.

Select ename from emp where Hiredate like '%FEB%';

realme Shot by SANDIP

(10) List the employees who are not working as manager and clerks in dept 10 and 20 with a salary in the range of 1000 to 3000.

Select ename from emp where job not in ('MANAGER', 'CLERK') AND deptno in (10,20) AND sal between 1000 AND 3000;

NOT LIKE: Not like operator is similar to LIKE operator instead of selecting the values it will reject the values.

Syntax:

Column-name / Expr NOT LIKE 'pattern-to-Match';

Emp wants NOT LIKE OR operator behave like 'AND' operator instead of OR

(1) WAQTD names of emps whose name doesn't starts with S.

Select ename from emp where ename NOT LIKE 'S%';

(2) WAQTD name and salary of emps whose name doesnot end with S.

Select ename, sal from emp where ename NOT LIKE '%S';

(3) WAQTD names of emps whose name doesnot have exactly 5 characters.

Select ename from emp where ename NOT LIKE '-----';

(4) WAQTD names of emps whose names doesnot starts with S or M.

→ Select ename from emp where ename NOT LIKE 'S%' AND NOT LIKE 'M%';

(5) WAQTD details of emps with their 7 month salary inc of 5% in their monthly salary and who are working in deptno 10, 20, 30, 40 and ~~not~~ working as president but emps are getting salary and reporting manager and hired in the range of

realme Shot by SANDY Realme 8i, 2024/02/11 12:34

1870 to 1890 and getting salary in the range between 10Rs to 10000 Rs and getting comm in the range of 0 to 5000 and names starts with vowels 'aeiou'.

→ ~~Select* from emp~~

Select emp.*, sal*7 + sal*7*5/100 from emp where deptno IN (10, 20, 30, 40) AND
job NOT IN 'PRESIDENT' AND
Mgr IS NOT NULL AND
~~(hiredate between 1870 AND 1890) AND~~
~~(Sal between 11 AND 9999) AND~~
~~(Comm between 0 AND 5000) AND~~
~~(ename LIKE 'A%' OR LIKE 'E%' OR LIKE 'I%' OR LIKE 'O%' OR LIKE 'U%');~~

(hiredate between '01-JAN-1870' and '31-dec-1890') and (Sal between 11 and 9999) and (comm between 1 and 4999) and
~~Ecom~~ is not null
(ename like 'a%' or ename like 'e%' or ename like 'i%' or ename like 'o%' or ename like 'u%');

realme Shot by ❤SANDIP❤
realme 8i 2024.02.11 12:18

Functions: It is a list of instruction of block of code which is used to perform some specific task is known as Functions.

Function
- In-built Func
  - Single Row Func (No. of I/P = No. of O/P)
  - Multi Row Func (No. of I/P = Single O/P)
- User-defined Func

> **Single Row Function:** ① If we pass single input to the single row function it will takes process it and gives the single O/P.

② If we pass multiple input to the single row function it will takes process it and gives the multiple output.

$$\boxed{\text{No. of I/P} = \text{No. of O/P}}$$

> **Multi Row Func:** ① If we pass multiple I/P to the multi row func it will takes process it and gives the single O/P.

② If we pass ~~multiple~~ single I/P to the multi row func it will takes process it and gives the single output.

$$\boxed{\text{No. of I/P} = \text{Single O/P}}$$

Multi Row func is also known as aggregate func / Group Func.

> Classified into 5 types:

i) MAX()
ii) MIN()
iii) SUM()
iv) AVG()
v) COUNT() [*]

realme Scooby ❤ SANDIP ❤
realme 8i 2024 02 11 12:18

① **MAX() Func:** It is used to ~~open~~ obtained the maximum value present in the Column.

② **MIN() Func:** It is used to ~~open~~ obtained the minimum value present in the Column.

③ **SUM() Func:** It is used to ~~open~~ obtained the addition or summesion of values Present in the Column.

④ **AVG() Func:** It is used to ~~open~~ obtained the average of values present in the Column.

⑤ **COUNT() Func:** It is used to ~~open~~ obtained the number of values present in the Column.

⑩ **NOTE:** ① Along with Multi Row Func it can accept a Column name or expression.

② MRF avoids null values.

③ MRF can pass only in SELECT clause.

④ Along with MRF it can't accept any other Column name.

⑤ COUNT() is the only MRF which can accept * as an arguments.

In where clause we can't pass Multi Row Function

① WAQTD maximum Sal of emps.

Select MAX(SAL) from emp;

②

EMP

EID	ENAME	SAL
1	A	100
2	B	200
3	C	300

SAL
100
200
300

O/P

MAX(SAL)
300

② WAQTD minimum Sal of emp.

→ Select MIN(SAL) from emp;

realme Shot by ❤SANDIP❤
realme 8i 2024 02 11 12:18

③ WAQTD max sal min sal of emps who are working as salesman.

Select max(sal), min(sal) from emp where job = 'SALESMAN';

④ WAQTD Avg sal and sum of sal of emps who are getting comm.

Select Avg(sal), Sum(sal) from emp where comm IS NOT null;

⑤ WAQTD number of emps.

Select COUNT(*) from emp;

⑥ WAQTD names of emps with their max sal.

~~Select ename, max(sal~~

Assignment of MAX & MIN

① WAQTD name of the employee earning maximum salary.

Select ename from emp where sal In (Select max(sal) from emp);

② WAQTD name of the employee earning minimum salary.

Select ename from emp where sal In (Select min(sal) from emp);

③ WAQTD name and hiredate of the employee hired before All the emps (first emp)

Select ename, hiredate from emp where hiredate In (Select min(hiredate) from emp);

④ WAQTD name hiredate of the employees hired at the last.

Select ename, hiredate from emp where hiredate In (select max (hiredate) from emp);

⑤ WAQTD name, comm of the employee who earns min comm.

Select ename, comm from emp where comm In (Select min (comm) from emp);

realme Shot by ❤SANDIP❤
realme 8i 2024 02 11 12:18

⑥ WAQTD name, sal and comm of the employee earning maximum comm.

```
Select ename, sal, comm from emp where comm In
(select max(comm) from emp);
```

⑦ WAQTD details of the emps who has greatest empno.

```
Select * from emp where empno In (Select max(empno)
from emp);
```

⑧ WAQTD details of the emps having the least hiredate.

```
Select * from emp where hiredate In (Select
min (hiredate) from emp);
```

⑨ WAQTD details of the emps earning least annual salary.

```
Select * from emp where sal*12 (Select min(sal*12)
from emp);
```

⑩ WAQTD name, annual salary of the emps if their annual salary is more than all the salesman.

```
Select ename, sal*12 from emp where sal*12>
All (Select sal*12 from emp where job
= 'SALESMAN');
```

① WAQTD no. of emps who are working in each dept.

```
Select count(*), deptno from emp group by
deptno;
```

Group By clause: Group-By clause is used to group the records.

Syntax:

```
SELECT GROUP.BY.F/ GROUP.BY EXPR.
FROM Table.Name
[where <filter-condition>]
GROUP BY Column Name;
```

optional

realme Shot by ❤SANDIP❤
realme 8i 2024 02 11 12:18

Order of execution:

FROM

optional ↳ Where - Row - By - Row

Group By - Row - By - Row

Select - Group - By - Group

o/p from clause

Eid	Ename	Sal	Dno
1	A	100	10
2	B	200	20
3	C	300	30
4	D	400	40
5	E	500	20
6	F	600	30
7	G	700	10
8	H	800	30
9	I	900	20
10	J	1000	20
11	K	100	30
12	L	200	20
13	M	100	40

o/p of group-by clause

10			
1	A	100	10
7	G	700	10

=2

20			
2	B	200	20
5	E	500	20
9	I	900	20
10	J	1000	20
12	L	100	20

=5

40			
4	D	400	40
13	M	300	40

=2

30			
3	C	300	30
6	F	600	30
8	H	800	30
11	K	200	30

=4

o/p of select clause

Count(*)	deptno
2	10
5	20
4	30
2	40

Select Count(*), deptno from emp group by dept no;

① WAQTD no. of emps who are working as Salesman, clerk, manager in each dept.

→ ~~Sect~~ Select count (*), deptno from emp where job in ('SALESMAN', 'CLERK', 'MANAGER') group by deptno;

② WAQTD no. of emps who are working in each job.

Select count(*), job from emp group by job;

realme Shot by SANDIP
realme 8i 2024 02 11 12:19

③ WAQTD maximum sal, minimum sal and no of emps who are working each dept if they are getting salary in the range of 1000 to 5000 and having reporting mgr.

Select ~~count(*), job~~, max(sal), min(sal)
~~from emp where j~~
max(sal), min(sal), count(*), deptno
from emp where (sal between 1000 and 5000) AND mgr is not null
group by deptno;

④ WAQTD avg sal, sum of sal of emps who are working in each job.

Select avg(sal), sum(sal), ~~count(*)~~, job
from emp group by job;

⑤ WAQTD no. of emps who are working in each job and getting same sal.

Select Count(*), job, sal from emp group by job, sal;

Note: ① Group-by clause is used to group the records

② Group-by clause executes row-by-row

③ without execution of where clause group-by clause will execute

④ After execution of group-by clause we get groups.

⑤ After group-by clause which ever clause will execute it will be executed in group-by group.

⑥ The column name which is passing in group by clause can be passed in select clause that is known group by expressions

realme Shot by ❤SANDIP❤

⑥ WAQTD no. of emps who are working in each dept their has to be atleast 4 employees.

→ Select count(*), deptno from emp group by deptno having count(*) ≥ 4;

HAVING CLAUSE: Having Clause is used to filter the groups.

Syntax:

```
Select group.by.F / group.by Expr
From Table_Name
where <filter-condition>
group-by column Name
Having <group-filter-Condition>;
```

optional

Note Points: ① After execution of group-by clause having clause will execute.

② Having clause executes group-by-group

③ Having clause can accept MRF.

④ Without where clause having clause can executes.

⑤

Order of execution:

optional

FROM
[where - Row - by - Row]
Group by - Row - by - Row
HAVING - Group - by - Group
Select - Group - by - Group

Q) WAQTD no. of emps who are working in each dept their has to be atleast 4 emps.

→ Select count(*), deptno from emp group by deptno having ~~clause~~ count(*) ≥ 4;

realme Shot by ❤SANDIP❤

O/P of emp clause

EID	Ename	Sal	DNo
1	A	100	10
2	B	200	20
3	C	300	30
4	D	400	40
5	E	500	20
6	F	600	30
7	G	700	10
8	H	800	30
9	I	900	20
10	J	100	20
11	K	200	30
12	L	100	20
13	M	300	40

O/P of group by clause

10

1	A	100	10
7	G	700	10

=2

20

2	B	200	20
5	E	500	20
9	I	900	20
10	J	100	20
12	L	100	20

=5

40

4	D	400	40
13	M	300	40

=2

30

3	C	300	30
6	F	600	30
8	H	800	30
11	K	200	30

=4

Count(*) ≥ 4

2 ≥ 4 X

5 ≥ 4 ✓

4 ≥ 4 ✓

2 ≥ 4 X

O/P of Having clause

20

2	B	200	20
5	E	500	20
9	I	900	20
10	J	100	20
12	L	100	20

=5

30

3	C	300	30
6	F	600	30
8	H	800	30
11	K	200	30

=4

O/P of Select Clause

Count(*)	DNo
5	20
4	30

① WAQTD no of emps, who are working in each job their has to be atleast.

Select count(*), job from emp
Group by job having count(*) ≥ 3;

② WAQTD max sal, min sal of emps who are working in each dept.
and their max sal exceeds of

realme
Shot on realme 8i 2024.02.11 12:19

3000 and min Sal exceeds of 5.

~~Select max(5), min(~~

Select Max(sal), min(sal), deptno from emp group by deptno having max(sal)> 3000 and min (sal)>5;

Assignment on MRFC

① WAQTD number of emps getting salary less than 2000 in deptno 10.

Select count(*) from emp where sal <2000 and deptno =10;

② WAQTD total sal needed to pay employees working as clerk.

Select sum(sal) from emp where job ='CLERK';

③ WAQTD avg sal needed to pay all employees.

Select avg(sal) from emp;

④ WAQTD number of emps having 'A' as their first character.

Select Count(*) from emp where ename like 'A%';

⑤ WAQTD number of employees working as clerk or manager.

Select Count(*) from emp where job='CLERK' OR job = 'MANAGER';

⑥ WAQTD total sal needed to pay employees hired in feb.

Select sum (sal) from emp where Hiredate like '%FEB%';

⑦ WAQTD number of emps reporting to 7839 (MGR)

Select count(*) from emp where mgr = 7839;

⑧ WAQTD num realme, emps getting [illegible] in deptno 30

Select count (*) from emp where comm is not null and deptno =30;

(9) WAQTD Avg Sal, total Sal, number of emps and maximum Sal given to emps working as President.

Select count(*), avg(sal), sum(sal), max(sal) from emp where job = 'PRESIDENT';

(10) WAQTD number of emps having 'A' in their names.

Select count(*) from emp where ename like '%A%';

(11) WAQTD number of emps and total sal needed to pay the emps who have 2 consicutive L's in their names.

Select count(*), sum(sal) from emp where ename like '%LL%';

(12) * WAQTD number of depts present in emp table.

Select count(distinct deptno) from emp;

(13) WAQTD number of emps having character 'Z' in their names.

Select count(*) from emp where ename like '%Z%';

(14) WAQTD number of emps having '$' in their names.

Select count(*) from emp where ename like '%$%';

(15) WAQTD total salary given to emps working as clerk in dept 30

Select sum(sal) from emp where job = 'CLERK' AND deptno = 30;

(16) WAQTD Maximum sal given to the emps working as Analyst.

Select max(sal) from emp where job = 'ANALYST';

realme Shot by ❤SANDIP❤

(17) ** WAQTD number of distinct salary present in emp table.

Select count (distinct sal) from emp;

(18) ** WAQTD number of jobs present in emp table.

Select count(distinct job) from emp;

(19) WAQTD Avg salary given to the clerk.

Select avg(sal) from emp where job='CLERK';

(20) WAQTD Minimum sal given to the emps who work in as manager or a clerk.

Select min(sal) from emp where job='MANAGER' OR job='CLERK';

Assignment Questions ON Group By clause

(1) WAQTD number of emps working in each Dept except president.

Select count(*), deptno from emp where job NOT IN ('PRESIDENT') group by deptno;

(2) WAQTD total salary needed to pay all the emps in each job.

Select sum(sal), job from emp group by job;

(3) WAQTD number of emps working as manager in each Dept.

Select count(*), deptno from emp where job='MANAGER' group by deptno;

(4) WAQTD avg salary needed to pay all the emps in each dept excluding the emps of deptno 20.

Select avg(sal), deptno from emp where deptno NOT IN 20 group by deptno;

(5) WAQTD number of emps having character 'A' in their names in each job.

Select count(*), job from emp where ename like '%A%' group by job;

realme Shot by SANDIP
realme 6i 2024 12 11 18:54

⑥ WAQTD number of emps and avg salary salary needed to pay the emps who salary in greater than 2000 in each dept.

Select count(*), avg(sal), deptno from emp where sal > 2000 group by deptno;

⑦ WAQTD total salary needed to pay and number of ~~salaries~~ salesman in each dept.

Select count(*), sum(sal), deptno, job from emp where job = 'SALESMAN' group by deptno, job;

⑧ WAQTD number of emps with their maximum salaries in each job.

Select count(*), max(sal), job from emp group by job;

⑨ WAQTD max sal given to an emp working in each dept.

Select max(sal), deptno from emp group by deptno;

⑩ WAQTD number of Times the salaries present in employee Table.

** Select count (distinct sal) from emp;

Assignment Questions ON HAVING CLAUSE

① WAQTD DNo and num of emp working in each Dept if there are atleast 2 clerks in each dept.

Select count(*), deptno, job from emp where job = 'CLERK' group by deptno, job Having count(*) >= 2;

realme Shot by ❤SANDIP❤

② WAQTD Dno and total sal needed to pay all emp in each dept if there are atleast 4 emp in each dept.

Select ~~(*)~~count(*), sum(sal), deptno from emp group by deptno Having count(*)≥4;

③ WAQTD number of emp earning sal more than 1200 in each job and the total sal needed to pay emp of each job must excess 3800.

~~Select job from emp~~
Select count(*), job from emp where sal > 1200 group by job having sum(sal) > 3800;

④ WAQTD deptno and number of emp working only if there are 2 Emp working in each dept as manager.

Select count(*), deptno from emp where job = 'MANAGER' group by deptno Having count(*) ≥ 2;

⑤ WAQTD job and max sal of emp in each job if the max sal exceeds 2600.

Select job, max(sal) from emp group by job having max(sal) > 2600;

⑥ WAQTD the salaries which are repeated in emp table.

Select sal from emp group by sal having count(*) ≥ 1;

⑦ WAQTD the hiredate which are duplicated in emp table.

Select hiredate from emp group by hiredate having count(*) ≥ 1;

⑧ WAQTD AVG sal of each dept if AVG sal is less than 3000.

Select avg(sal), deptno from emp group by deptno having avg(sal) < 3000;

realme Shot by ❤SANDIP❤

(9) WAQTD deptno if there are atleast 3 emp in each Dept whos name has char 'A' or 'S'

Select deptno from emp where ename like '%A%' OR ename like '%S%' Having count(*)>3;

(10) WAQTD MIN AND MAX Salaries of each job if min Sal is more than 1000 and Max Sal is less than 5000.

Select min(Sal), max(Sal), job from emp group by job Having min(Sal) > 1000 AND max(Sal) < 5000;

Order By clause: ① Order-By clause is used to arrange the records in ascending or Descending order.

② If the records have to be arrange in ascending order we have to pass the Keyword of 'ASC'

③ without passing Keyword of 'ASC' the compiler will automatically arrange the records in ascending order.

④ If the Column of records want be executed in desending order we have to pass the Keyword of 'desc' it is mandatory.

Syntax:

Select group-By-exp/ Group By Func
From Table-name
[where <filter-condition>
group by Column-name
Having <group-filter condition>]
Order By Column Name;

Optional

⑦ Order of execution:

From
Where - Row-by-Row
Group By - Row-by-Row
Having - Group-by - Group
Select - Group-by-Group
OrderBy - Row-By-Row

① WAQTD names of emps arrange the records in asc order.

→ Select ename from emp ~~group~~ order by ename asc;

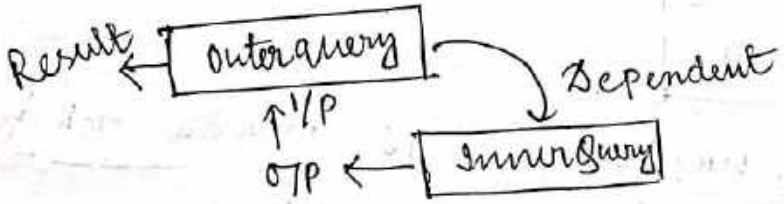
② WAQTD job of emps who are working as Salesman, Manager, Analyst and in each job their has to be atleast 1 employee an arrange the records of job in desc.

→ Select job from emp where job = 'SALESMAN' OR job = 'MANAGER' OR job = 'ANALYST' group by job having count(*) > 1 order by job desc;

③ WAQTD names of emps who are getting Salary more than jones.

Select ename from emp where sal > (select sal from emp where ename = 'JONES');

④ Sub-Query: A Query written inside another Query is known as Subquery.



realme Shot by ❤SANDIP❤

Working Process of Sub-Query:

① Let us, consider two queries outer queries and inner query.

② 1st, Inner query will execute and Produces the output.

③ The output of inner query is passed as input to the outer query

④ By taking this input outer query will produces the result.

⑤ So, we can state that, outer queries dependend on inner query.

Case-I: ① Whenever there is an unknown Present in the query we used Subquery to find the unknown.

① WAQTD name of emp getting sal > jones.

```
Slect ename
 from emp
 where sal > (Select sal
              from emp
              where ename='JONES');
```

Emp table:

EID	Ename	SAL
1	SMITH	100
2	FORD	250
3	JONES	170
4	KING	200
5	MILLER	150
6	WARD	190

O/P table:

EName
FORD
KING
WARD

② WAQTD names of emps who are getting sal < Miller.

```
Select ename from emp
 where sal < (Select sal from emp
              where ename = 'MILLER');
```

realme Shot by ❤SANDIP❤
realme 8i 2024 02 11 12:34

③ WAQTD details of emp who are hiredate before JAMES.

Select * from emp where ~~ename~~ hiredate < ~~where~~ (Select ~~*~~ ~~from~~ hiredate from emp where ~~ename~~ ename = 'JAMES');

④ WAQTD name, sal, job of emps who are getting sal more than MARTIN and working as Manager.

Select ename, sal, job from emp where sal > (Select ~~job~~ ~~ename~~ sal from emp where ename = 'MARTIN') AND job = 'MANAGER';

⑤ WAQTD name and hiredate of emps who are hired after allen before King.

Select ename, hiredate from emp where hiredate > (Select hiredate from emp where ename = 'ALLEN') AND hiredate < (Select hiredate from emp where ename = 'KING');

⑥ WAQTD names of emps who are working in the same job as WARD.

Select ename from emp where job = (Select ~~ename~~ job from emp where ~~job~~ ename = 'WARD');

realme Shot by ❤SANDIP❤
realme 8i 2024.02.11 12:34

Assignment on case-I

① WAQTD Number of emps getting Salary less than ~~2000 in deptno 10.~~

② WAQTD ~~total~~ Sal ~~needed to pay~~ emps ~~working as clerk.~~

③ WAQTD ~~avg~~ Sal ~~needed~~

① WAQTD name of the emps earning more than ADMAS.

Select ename from emp where sal > (Select sal from emp where ename = 'ADMAS');

② WAQTD name and Sal of the emps earning less than KING.

Select ename, sal from emp where sal < (Select sal from emp where ename = 'KING');

③ WAQTD ~~ename~~ and deptno of the emps if they * are working in the same dept as JONES.

Select ename, deptno from emp where deptno = (Select deptno from emp where ename = 'JONES');

④ WAQTD ~~empno~~ name and job all the emps working in the same designation As James.

Select ename, job from emp where job = (Select job from emp where ename = 'JAMES');

⑤ WAQTD empno and ename along with annual Salary of all the emps if their annual sal is greater than Wards annual sal.

Select empno, ename, sal*12 from emp where sal*12>
(Select sal*12 from emp where ename='WARD');

⑥ WAQTD name and hiredate of the emps if they are hired before SCOTT.

Select ename, hiredate from emp where hiredate
< (Select hiredate from emp where ename='SCOTT');

⑦ WAQTD name and hiredate of the emps if they are hired after the president.

Select ename, hiredate from emp where hiredate
> (Select hiredate from emp where job='PRESIDENT');

⑧ WAQTD name and sal of the emps if they are earning sal less than the employee whos empno is 7839.

Select ename, sal from emp where sal < (Select
sal from emp where empno=7839);

⑨ WAQTD all the details of the emps if the emps are hired before miller.

Select * from emp where hiredate < (Select hiredate
from emp where ename='MILLER');

⑩ WAQTD ename and empno of the emps if employees are earning more than ALLEN.

Select ename, empno from emp where sal>
(Select sal from emp where ename='ALLEN');

⑪ WAQTD ename and sal of all the emps who are earning more than MILLER but less than ALLEN.

Select ename, sal from emp sal> (Select
sal from emp where ename='MILLER')
And sal < (Select sal from emp where
ename='ALLEN');

⑫ WAQTD all the details of the emps working in dept 20 And working in same designation as Smith.

realme Shot by ❤SANDIP❤

Select * from emp where deptno = 20 and job = (Select job from emp where ename = 'SMITH');

(13) WAQTD all the details the emps working as manager in the same dept as turner.

Select * from emp where job = 'MANAGER' and deptno = (Select deptno from emp where ename = 'TURNER');

(14) WAQTD name and hiredate of the emps hired after 1980 and before King.

Select ename, hiredate from emp where hiredate > '31-DEC-1980' and hiredate < (Select hiredate from emp where ename = 'KING');

(15) WAQTD name and sal along with annual sal for all emps whos sal is less than Blake and more than 3500.

Select ename, sal, sal * 12 from emp where sal < (Select sal from emp where ename = 'BLAKE') and sal > ~~(Sele~~ 3500;

(16) WAQTD all the details of emps who earn more than Scott but less than KING.

Select * from emp where sal > (Select sal from emp where ename = 'SCOTT') and sal < (Select sal from emp where ename = 'KING');

(17) WAQTD name of the emps whos name starts with 'A' and works in the same dept as Blake.

Select ename from emp where ename like 'A%' and deptno = (Select deptno from emp where ename = 'BLAKE');

(18) WAQTD name and comm if emps earn comm and work in the same designation as Smith

Select ename, comm where comm is not null and job = (Select job from emp where ename = 'SMITH');

realme Shot by ❤SANDIP❤

(19) WAQTD details of all the emps working as clerk in the same dept as turner.

Select * from emp where job='CLERK' and deptno = (Select deptno from emp where ename = 'TURNER');

(20) WAQTD ename, sal and designation of the emps whos annual salary is more than smith and less than king.

Select ename, sal, job from emp where sal*12 > (Select sal*12 from emp where ename = 'SMITH') and sal*12 < (Select sal*12 from emp where ename = 'KING');

(21) WAQTD deptname of smith.

Select deptname from dept where dno = (Select dno from emp where ename = 'SMITH');

Sub-Query:

(i) Case-II: whenever, the data to be selected and condition to be executed are present in two different table we use subquery case-II.

(1) WAQTD deptname of smith.

Select dname from dept where dno = (Select dno from emp where ename = 'SMITH');

Emp

EID	Ename	DNO
1	SMITH	10
2	MILLER	20
3	KING	30
4	SCOTT	30

Dept

Dno	Dname	Loc
10	D1	L1
20	D2	L2
30	D3	L3

O/P:

Dname
D1

realme Shot by SANDIP

② WAQTD Location of allen.

Select loc from dept where deptno = (Select deptno from emp where ename = 'ALLEN');

③ WAQTD names of emps who are working in loc newyork.

Select ename from emp where ~~loc~~ deptno = (Select ~~loc~~ deptno from dept where loc = 'NEWYORK');

④ WAQTD names of emps who are getting Sal less than King and working in the loc dallas.

(LESS I + LESS II)

Select ename from emp where sal < (Select sal from emp where ename = 'KING') and deptno = (Select deptno from dept where loc = 'DALLAS');

⑤ WAQTD details of emps whose names contains character 'A' ~~and~~ and who are hired after smith and working in NEWyork.

Select * from emp where ename like '%A%' and hiredate > (Select hiredate from emp where ename = 'SMITH') and deptno = (Select deptno from dept where loc = 'NEWYORK');

⑥ WAQTD no of emps who are working in each dept and their has to be at least three peoples in it and and who are getting sal and not having comm but having reporting mgr and working in deptno 10, 20, 30, 40 and emps are hired before adams and working in chikago.

Select count(*), deptno from emp group by deptno
~~having count(*) >= 3~~ and
where sal is not null and comm is not null
and mgr is not null and deptno in (10, 20, 30, 10)
and hiredate < (Select hiredate from emp where
ename = 'ADMAS') and deptno = (Select
deptno from dept where LOC = 'CHICAGO');
group by deptno having count(*) >= 3;

⑦ WAQTD names of emps who are working in the loc newyork, Chikago.

Select ename from emp where deptno = (
Select deptno from dept where Loc
= 'NEWYORK' or Loc = 'CHIKAGO');

Types of Subquery: Two Types of Subquery
— ① Single Row Subquery
— ② Multi Row Subquery

① Single Row Subquery: ① If the subquery returns exactly one value or one record we call it as single row subquery.

② If the subquery returns one value we can use normal operators or special operators to comapre the values

① WAQTD names of emps who are working in dallas.

Select ename from emp where deptno = (Select
deptno from dept where Loc = 'DALLAS');

② Multi Row Subquery: ① If the subquery returns more than one value or one record we call it as multiRow subquery.

② If the subquery returns more than one value we can't use normal operators to compare the values instead we

realme Shot by ❤SANDIP❤

have to use Special operators to compare to values.

① Select dname from ~~dept~~ dname where ~~deptno~~ dno In (select ~~deptno~~ dno from emp where ename = 'SMITH');

Dept Table

Dno	Dname	loc
10	D1	L1
20	D2	L2
30	D3	L3

Emp Table

EId	Ename	DNO
1	SMITH	10
2	MILLER	20
3	SMITH	30

① WAQTD dname of emps who are working as salesman, manager.

Select dname from dept where deptno In (Select deptno from emp where job = 'SALESMAN' OR job = 'MANAGER'); job IN ('SALESMAN', 'MANAGER'));

Note:

1. We cannot define wheather a subquery belongs to single row / multi Row it is better to use special operator to compare the values of subquery at RHS.

② WAQTD names of emps who are working in Chikago and their name has character 'S' in it.

Select ename from emp where ename like '% S%' ~~where~~ AND deptno In (Select deptno from ~~emp~~ dept where loc = 'CHIKAGO');

③ WAQTD loc of emps whos name has char 'A' in it.

Select loc from dept where ~~ename like '%A%'~~ deptno In (select deptno from emp where ename like '%A%');

④ WAQTD details of emps with their 4 month sal and increament of 10% in their monthly sal and who are getting comm less than miller and having reporting manager and working in sales deptman and getting sal in the range between 1000 to 4000 and hired in the range of 1960 to 1990.

Select emp.*, sal*4, sal+sal*10/100 from emp where comm is not null and comm (select comm from emp where ename = 'MILLER') and mgr is not null and deptno = (select deptno from dept where dname = 'SALES') and (sal between 1001 and 3999) and (hire date between '01-JAN-1960 and '31-DEC-1990');

Assignment on Case 2

① WAQTD dname of the emps whos name is smith.

Select dname from dept where deptno = (select deptno from emp where ename = 'SMITH');

② WAQTD dname and loc of the employee whos ename is King.

Select dname, loc from dept where deptno = (select deptno from emp where ename = 'KING');

③ WAQTD loc of the emp whos employee number is 7902

Select loc from dept where deptno = (select deptno from emp where empno = 7902);

④ WAQTD dname and loc along with deptno of the emp whos name ends with 'R'.

Select dname, loc, deptno from dept where deptno In (select deptno from emp where ename like '%R');

realme Shot by ❤SANDIP❤

(5) WAQTD dname of the employee whos designation is president.

Select dname from dept where deptno In (Select deptno from emp where job = 'PRESIDENT');

(6) WAQTD names of the employee working in Accounting department.

Select ename from emp where deptno In (Select deptno ~~where~~ from dept where dname = 'ACCOUNTING')

(7) WAQTD ename and salaries of the employees who are working in the location Chicago.

Select ename, sal from emp where deptno In (Select deptno from dept where loc = 'CHICAGO');

(8) WAQTD details of the emps working in Sales.

Select * from emp where deptno ~~from~~ In (Select deptno from dept where dname = 'SALES');

(9) WAQTD details of the emp along with annual sal if employees are working in New York.

Select emp.*, sal*12 from emp where deptno In (Select deptno from dept where loc = 'NEWYORK');

(10) WAQTD names of employees working in operations department.

Select ename from emp where deptno In (Select deptno from dept where dname = 'OPERATIONS');

Assignment on Case 1 & 2

(1) WAQTD names of the employees earning more than scott in accounting dept.

Select ename from emp where sal > (Select sal from emp where ename = 'SCOTT') and deptno In (Select deptno from dept where dname = 'ACCOUNTING');

(2) WAQTD details of the emps working as manager in the location Chicago.

Select * from emp where job = 'MANAGER' and deptno In (Select deptno from dept where loc = 'CHICAGO');

realme Shot by ❤SANDIP❤

③ WAQTD name and sal of the emps earning more than King in the dept accounting.

Select ename, sal from emp where sal > (Select sal from emp where ename = 'KING') and deptno In (Select deptno from dept where dname = 'ACCOUNTING');

④ WAQTD details of the emps working as Salesman in the department Sales.

Select * from emp where job = 'SALESMAN' AND deptno In (Select deptno from dept where dname = 'SALES');

⑤ WAQTD name, sal, job, hiredate of the emps working in operations department and hired before king

Select ename, sal, job, hiredate from emp where deptno In (Select deptno from dept where dname = 'OPERATIONS') and hiredate < (Select hiredate from emp where ename = 'KING');

⑥ Display all the emps whos department names ending 's'.

* ~~Select * from dept where dname like '%s';~~

Select ename from emp where deptno In (Select deptno from dept where dname like '%S');

⑦ WAQTD dname of the emps whos names has character 'A' in it.

Select dname from dept where deptno In (Select deptno from emp where ename like '%A%');

⑧ WAQTD dname and loc of the emps whos Salary is rupees 800.

Select dname, loc from dept where deptno In (Select deptno from emp where sal = 800);

⑨ WAQTD dname of the employees who earn commission.

Select dname from dept where deptno In (Select deptno from emp where comm is not null);

⑩ WAQTD loc of the emps if they earn commission in dept 40.

Select loc from dept where deptno In (Select deptno from emp where comm is not null) and deptno In 40;

realme Shot by SANDIP
realme 8i 2024.02.11 12:25

Max and Min

Select ename from emp where sal = (Select max(sal) from emp);

Sub Query operator

① WAQTD names of emps who are getting sal more than all emps of deptno 10.

→ Select ename from emp where sal (All) (select sal from emp where deptno in 10);

Sub Query operators:

(i) All operator: ① All operator is a special operator which will accept along with relational operators (>, <, >=, <=) to compare the values at RHS.

** Relational operator cannot accept multivalued at RHS

② All operator returns true if all the values are RHS have satisfied the condition.

Emp

Eid	Ename	Sal	Dno
1	A	100	10
2	B	200	20
3	C	300	30
4	D	400	10
5	E	500	20
6	F	600	10

Dept no = 10

No Rows selected

100 > All 100 400 600
200 > All 100 400 600
300 > All 100 400 600
400 > All 100 400 600
500 > All 100 400 600
600 > All 100 400 600

(ii) Any operator: Any operator is a subquery operator which will accept along with relational operator to compare the values at RHS

② Any operator returns true if any one value have satisfied the condition.

① WAQTD names of the emps who are getting sal ~~from any of the~~ emp wherein deptno 10.

Select ename from emp where sal (Any) (select sal from emp where deptno in 10);

realme shot by SANDIP

① WAQTD name, sal of emps who are getting sal more than any one of the manager.

Select ename, sal from emp where sal > Any (select sal from emp where job = 'MANAGER');

② WAQTD max sal of emps.

Select max(sal) from emp;

③ WAQTD 2nd max sal of emps.

→ Select max (sal) from emp where sal < (select max (sal) from emp);

> Select Max(sal) from emp where sal < (select max (sal) from emp);

Eid	Ename	Sal
1	A	100
2	B	200
3	C	300
4	D	400
5	E	500
6	F	600

100 < 600
200 < 600
300 < 600
400 < 600
500 < 600 2nd max sal
600 < 600 ✗

Nested Subquery: ① A sub query written inside another subquery is known as Nested Subquery.

② We can nested upto 255 sub query.

① WAQTD 3rd max sal of emps

Select max (sal) from emp where sal < (select max(sal) from emp where sal < (select max (sal) from emp));

② WAQTD 2nd min sal of emps

Select min (sal) from emp where sal > (select min (sal) from emp);

realme Shot by ❤SANDIP❤
realme 8i 2024 02 11 12:35

③ WAQTD names of emps who are getting 2nd max sal.

Select ; ename from emp where sal <
~~(Select max(sal) from emp)~~;

=(Select max (sal) from emp where
sal < (Select max(sal) from emp));

④ WAQTD details of emps who are getting 2nd max sal.

Select * from emp where sal In
(Select max (sal) from emp where sal<
(Select max (sal) from emp));

⑤ WAQTD names, sal of emps who are getting 2nd minimum sal.

Select ename, sal from emp where sal In
(Select min (sal) from emp where sal>
(Select min (sal) from emp));

⑥ WAQTD names of emps who are hired 1st.

Select ename from emp where hiredate
In (Select min(hiredate) from emp);

⑦ WAQTD name and hiredate of emps who are hired 3rd

Select ename, hiredate from emp where
hiredate In (Select min(hiredate) from
emp where hiredate > (Select min (hiredate) from
emp where hiredate > (Select min
(hiredate) from emp)));

realme Shot by SANDIP

Assignment on Types of Sub-Query

1. WAQTD name of the emp earning sal more than the salesman

Select ename from emp where sal > All (Select sal from emp where job = 'SALESMAN');

(2) WAQTD details of the emps hired after all the clerks.

Select * from emp where hiredate > All (Select hiredate from emp where job = 'CLERK');

(3) WAQTD name and sal for the all emps if they are earning less than atleast a manager.

Select ename, sal from emp where sal < Any (Select sal from emp where job = 'MANAGER');

(4) WAQTD name and hiredate ~~for all~~ of emps hired before all the managers.

Select ename, hiredate from emp where hiredate < All (Select hiredate from emp where job = 'MANAGER');

(5) WAQTD names of the emps hired after all the managers and earning salary more than all the clerks.

Select ename from emp where hiredate > All (Select hiredate from emp where job = 'MANAGER') And sal > All (Select sal from emp where job = 'CLERK');

(6) WAQTD details of the emps working as clerk and hired before atleast a salesman

Select * from emp where job = 'CLERK' and hiredate < Any (Select hiredate from emp where job = 'SALESMAN');

(7) WAQTD details of the emps working in Accounting or sales dept.

Select * from emp where deptno In (Select deptno from dept where dname = 'ACCOUNTING') or deptno In (Select deptno from dept where dname = 'SALES');

realme Shot by ❤SANDIP❤
realme 8i 2024 02 11 12:36

(8) WAQTD department names of the emps with names Smith, King, Miller.

Select dname from dept where deptno In (Select deptno from emp where ename='SMITH') or deptno In (Select deptno from emp where ename='KING') or deptno In (Select deptno from emp where ename='MILLER');

(9) WAQTD Details of emps looking Newyork or Chicago.

Select * from emp where deptno In (Select deptno from dept where loc='NEWYORK') or deptno In (Select deptno from dept where loc='CHICAGO');

(10) WAQTD emp names if emps are hired after all the emps of dept 10.

Select ename from emp where hiredate > All (Select hiredate from emp where deptno=10);

Assignment on Nested SubQuery

(1) WAQTD 2nd min Sal.

Select min(sal) from emp where sal > (Select min (sal) from emp);

(2) WAQTD 5th maximum Sal.

Select max(sal) from emp where sal < (Select max(sal) from emp where sal < (Select max(sal) from emp where sal < (Select max(sal) from emp))));

(3) WAQTD Name of the emps earning 3RD max Sal.

~~Select max (sal) from emp where sal < (Select max(sal) from emp where sal < (Select max(sal) from emp where));~~

Select ename from emp where sal In (Select max(sal) from emp where sal < (Select max(sal) from emp where sal < (Select max (sal) from emp)));

(4) WAQTD empno of the employee earning 2nd max Sal.

Select empno ~~max(sal)~~ from emp where ~~sal < (Select max(sal) from emp)~~, sal In (Select max(sal) from emp where sal < (Select max (sal) from emp));

⑤ WAQTD dept name of an employee getting 4th max sal.
Select dname from dept where deptno In (select deptno from emp where sal In (select max (sal) from emp where sal< (select max (sal) from emp where sal< (select max(sal) from emp where sal< (select max (sal) from emp)))));

⑥ WAQTD details of the emps who was hired 2nd.
Select * from emp where hiredate In (select min (hiredate) from emp where hiredate> (select min (hiredate) from emp));

⑦ WAQTD name of the employee hired before the last employee.
Select ename from emp where hiredate ~~In~~ (select ~~for~~ max (hiredate) from emp where hiredate < (select max (hiredate) from emp));

⑧ WAQTD loc of the emp who was hired first.
Select loc from dept where deptno In (select deptno from emp where hiredate> (select min (hiredate) from emp));

⑨ WAQTD Details of the emps earning 7th minimum sal.
Select * from emp where sal In (select min (sal) from emp where sal> (select min (sal) from emp where sal> (select min (sal) from emp where sal> (select min (sal) from emp where sal> (select min (sal) from emp where sal> (select min (sal) from emp)))))));

⑩ WAQTD dname of emp getting 2nd maximum sal.
~~Select dname, min(sal) from emp where sal< (select max (sal) from emp);~~
Select dname from dept where deptno In (select deptno from emp where sal In (select max (sal) from emp where sal< (select max (sal) from emp));

realme Shot by ❤SANDIP❤

Clustered Index: A clustered index in SQL is a type of index that determines the physical order of data in a table. It defines the order in which the rows of a table are stored on disk or other storage media.

A clustered index is created on a unique identifier column, such as a primary key and significantly impacts the table's storage and query performance. A table can have only one clustered index and it is automatically created when a primary key or a unique constraints is defined on the table.

Non-clustered Index: A non-clustered index in SQL is a type of index that does not determine the physical order of data in a table, but creates a separate structure that contains the column-values and pointers to the data rows.

A non-clustered index can improve the query performance by finding the matching rows faster than scanning the whole table.

A table can have multiple non-clustered indexes, up to 999 per table.

① WAQTD Smith's reporting Manager name.

Employee Manager Relation:

i) To find Manager

$\boxed{MGR = EMPNO}$

EId	Ename	MGR
1	ALLEN	3
2	JONES	1
3	SMITH	2
4	KING	4

Select ename from emp where empno = (Select mgr from emp where ename = 'SMITH');

② WAQTD details of Adams reporting manager

Select * from emp where empno = (Select mgr from emp where ename = 'ADAMS');

③ WAQTD names of employees who are reporting to Ford.

Select ename from emp where ~~empno~~ mgr In (Select ~~mgr~~ empno from emp where ename = 'FORD');

ii) To Find Employees

$\boxed{Empno = Mgr}$

④ WAQTD names of employees who are reporting to BLAKE.

Select ename from emp where mgr In (Select empno from emp where ename = 'BLAKE');

⑤ WAQTD Smith's manager's manager name.

Select ename from emp where empno ~~In (Select mgr from emp where ename = 'SMITH') And~~ where empno = (Select ~~Empno~~ mgr from emp where ename = 'SMITH');

⑥ WAQTD details of Adams manager's manager name.

Select * from emp where empno In (Select mgr from emp where empno In (Select mgr from emp where ename='ADAMS'));

⑦ WAQTD names of emps who are reporting to Scott manager

⇒ Select ename from emp where mgr In (Select empno from emp where ~~ename='SCOTT'~~); empno In (Select mgr from emp where ename='SCOTT'));

Another method ⇒ Select ename from emp where mgr In (Select mgr from emp where ename='SCOTT');

⑧ WAQTD details of emps who are reporting to Smith's manager.

Select* from emp where mgr In (Select mgr from emp where ename='SMITH');

⑨ WAQTD details of emps with their 9 month salary decrement of 1 rupee in their daily sal and emps are getting sal more than ~~sal~~ Smith and hired after Smith and working in the loc newyork, Chicago, dollas and getting 3rd max sal and reporting King.

~~Select*~~ from Select emp.*, sal*9, sal/30 - 1 ~~and~~ from emp where sal > (Select sal from emp where ename='SMITH') And hiredate > (Select hiredate from emp where ename='SMITH') and deptno = (Select deptno from dept where loc In('NEWYORK', 'CHICAGO', 'DOLLAS')) and ~~sal In (Select max(sal) from emp where~~

Sal (Select max(sal) from emp where sal < (Select max(sal) from emp where sal < (Select max(sal) from emp))) where mgr In (Select empno from emp where ename = 'KING');

Assignment on Emp and Manager Relation

① WAQTD Smith's reporting manager's name.
Select ename from emp where empno In (Select mgr from emp where ename = 'SMITH');

② WAQTD Adams manager's manager name.
Select ename from emp where empno In (Select mgr from emp where empno In (Select mgr from emp where ename = 'ADAMS'));

③ ** WAQTD dname of Jones manager.
Select dname from dept where deptno In (Select deptno from emp where empno In (Select mgr from emp where ename = 'JONES'));

④ WAQTD Miller's manager's salary.
Select sal from emp where empno In (Select mgr from emp where ename = 'MILLER');

⑤ WAQTD loc of Smith's manager's manager.
Select loc from dept where deptno In (Select deptno from emp where empno In (Select mgr from emp where empno In (Select mgr from emp where ename = 'SMITH'));

⑥ WAQTD name of the employees reporting to Blake.
Select ename from emp where mgr In (Select empno from emp where ename = 'BLAKE');

⑦ WAQTD number of employees reporting to king
Select count (*) from emp where mgr In (Select empno from emp where ename = 'KING');

⑧ WAQTD details of the employees reporting to ~~Blake's manager~~. Jones.
Select * from emp where mgr In (Select empno from emp where ename = 'JONES');

(9) WAQTD enames of the employees reporting to Blake's manager.

Select ename from emp where mgr In (Select mgr from emp where ename = 'BLAKE');

(10) WAQTD number of employees reporting to Ford's manger.

Select Count(*) from emp where mgr In (Select mgr from emp where ename = 'FORD');

Date - 17.08.23

JOINS

It is used to retrieve / fetch the data from multiple table simultaneously is known as JOINS

> Joins are classified into 5 types.

1. Cross Join / Cartesian Join
2. Inner Join / Equi Join
3. Outer Join
   - → Left Outer Join
   - → Right outer Join
   - → ~~Self~~ Full Outer Join
4. Natural Join
5. Self Join

Cross / Cartesian Join: The records of table1 will be must with the all the records of Table2.

① WAQTD ename and dname of employees.

Ename	DNo
A	10
B	20
C	30

DName	DNo
D1	10
D2	20
D3	30

Ename	DNo	DName	DNo
A	10	D1	10
A	10	D2	20
A	10	D3	30
B	20	D1	10
B	20	D2	20
B	20	D3	30
C	30	D1	10
C	30	D2	20
C	30	D3	30

Syntax:

ORACLE:

Select Column_Name from Table-Name1, Table_Name2;

ANSI (American National Standard Institute)

Select Column_Name from Table-Name1 CROSS JOIN Table-Name2;

① WAQTD ename, dname of employees

→ Select ename, dname from emp, dept;

→ Select ename, dname from emp CrossJoin dept;

Inner Join / Equi Join: It is used to obtained exactly matched the records.

> WAQTD ename and dname of emps

Syntax:

> Oracle: Select Column-Name from Table-name1, tableName2 [where <Join-Condition>];

> ANSI:

Select Column_Name from Table-name1 [INNER JOIN] Table-Name2 [ON] [<Join-Condition>];

Emp

Ename	DNo
A	10
B	20
C	30

Dept

DName	DNo
D1	10
D2	20
D3	30

Ename	DNo	DName	DNo
A	10	D1	10
B	20	D2	20
C	30	D3	30

→ Select ename, dname from emp, dept where emp.dno = dept.dno;

→ Select ename, dname from emp Inner Join Dept on emp.dno = dept.dno;

① Name of the employee and his location of all the employees.

Select ~~emp~~ ename, loc from emp, dept where emp.deptno = dept.deptno;

② WAQTD dname and salary for all the employee working in Accounting.

Select dname, sal from emp, dept where dept.deptno = emp.deptno and ~~job~~ dname = 'ACCOUNTING';

③ WAQTD dname and annual salary for all emps who salary is more than 2340.

Select dname, sal*12 from emp, dept where emp.deptno = dept.deptno and sal > 2340;

④ WAQTD dname and annual salary for all employees whos salary is more than 2340.

Select dname, sal*12 from emp, dept where emp.deptno = dept.deptno and sal > 2340

⑤ WAQTD ename and dname for employees having character 'A' in their dname.

Select ename, dname from emp, dept where emp.deptno = dept.deptno and dname like '%A%';

⑥ WAQTD ename and dname for all the emps working as salesman.

Select dname, ename from emp, dept where dept.deptno = emp.deptno and job = 'SALESMAN';

⑦ WAQTD dname and job for for all the employees whos job and dname starts with character 'S':

Select dname, job from emp, dept where emp.deptno = emp.deptno and job like '% S%' and dname like '% S%';

⑧ WAQTD dname and mgr no for employees reporting to 7839.

Select dname, mgr from emp, dept where dept.deptno = emp.deptno and mgr = 7839;

⑨ WAQTD dname and hiredate for employees hired after 83 into Accounting or Research dept.

Select dname, hiredate from emp, dept where dept.deptno = emp.deptno and hiredate > '31-DEC-1983' and (dname = 'ACCOUNTING' or dname = 'RESEARCH');

⑩ ** WAQTD ename and dname of the employees who are getting comm in dept 10 or 30.

Select ename, dname from emp, dept where dept.deptno = emp.deptno and emp.deptno In (10, 30) and Comm is not null;

⑪ WAQTD dname and empno for all the emps whos empno are (7839, 7902) and are working in loc newyork.

Select dname, empno from emp, dept where dept.deptno = emp.deptno and empno In (7839, 7902) and loc = 'NEW YORK';

① ** WAQTD ename and dname of emps who are getting sal more than Allen.

Select ename, dname from emp, dept where emp.deptno = dept.deptno and Sal > (Select Sal from emp where ename = 'ALLEN');

Outer Join: It is used to ~~open~~ obtained the unmatched records.

i) Left Outer Join: It is used to ~~open~~ obtained the unmatched records of left table along with matched records.

① WAQTD ename and dname of emps even though their are no emps working in a dept.

Emp:

Ename	Dno
A	10
B	
C	30
D	

(L)

Dept:

Dname	Dno
D1	10
D2	20
D3	30
D4	40

(R)

M, UM

Result Table

Ename	Dno	Dname	Dno
A	10	D1	10
C	30	D2	20
B	NULL	NULL	NULL
D	NULL	NULL	NULL

Ename	Dname
A	D1
C	D2
B	NULL
D	NULL

Syntax:

➤ **ORACLE:** Select Column Name from TN1, TN2 where <Join Condition>;
emp.deptno = dept.deptno(+);

Select ename, dname from emp, dept where emp.deptno = dept.deptno (+);

➤ **ANSI:** Select ColumnName from TN1 Left [outer] Join TN2 on <Join-Condition>;

Select ename, dname from emp Left Join dept on emp.deptno = dept.deptno;

ii) **Right Outer Join:** It is used to obtained the unmatched records of right table along with matched records.

① WAQTD ename and dname of emps even through there are no emps in a dept.

Emp:

	Ename	Dno
M	A	10
	B	
UM	C	30
	D	

(L)

Dept:

	Dname	Dno
M	D1	10
	D2	20
UM	D3	30
	D4	40

(R)

Result Table

Ename	Dno	Dname	Dno
A	10	D1	10
C	30	D3	30
NULL	NULL	D2	20
NULL	NULL	D4	40

Ename	Dname
A	D1
C	D3
NULL	D2
NULL	D4

> Syntax:

Oracle:

Select Column_Name from TN1, TN2 where <Join Condition>;

emp. deptno (+) = dept. deptno;

ANSI:

Select Column_Name from TN1 Right [Outer] Join TN2 on <Join-Condition>;

⟹ Select ~~emp~~ ename, ~~dept~~ dname from emp, dept where emp. deptno (+) = dept. deptno;

⟹ Select ename, dname from ~~emp, dept~~ Right join dept on emp. deptno = dept. deptno;

iii) **Full outer Join:** It is used to obtained the unmatched and matched records of both the tables.

① WAQTD ename, dname of emps even though the emps doesnot work in any dept and a dept has no employees.

Emp:

Ename	Dno
A	10
B	
C	30
D	

(L)

Dept:

Dname	Dno
D_1	10
D_2	20
D_3	30
D_4	40

(R)

Ename	Dno	Dname	Dno
A	10	D_1	10
C	30	D_3	30
NULL	NULL	D_2	20
NULL	NULL	D_4	40
B	NULL	NULL	NULL
D	NULL	NULL	NULL

ANS1: Select Column Name from TN_1, Full[outer] Join TN_2 on <Join-Condition>;

→ Select ename, dname from emp Full join dept on emp.deptno = dept.deptno;

Natural Join: Natural Join acts in two ways:

① **Case-I:** whenever there is connection betn two tables it acts like inner join.

② **Case-II:** whenever there is no connection betn two tables it acts like cartession or cross join.

> Syntax:

ANSI: Select Col-name from TN1 Natural Join TN2;

Self Join: Joining the table by itself is known as Self Join.

→ Why or where do you use Self Join: whenever the data's to be selected are present in same table but in different records

① WAQTD emp name and mgr/manager name of employees.

→ Select E1.Ename, E2.Ename from emp E1, emp E2 where E1.MGR = E2.EId;

→ Select E1.Ename, E2.Ename from emp E1 Join emp E2 on E1.MGR = E2.EId;

EMP E1

E1.EId	E1.Ename	E1.MGR
1	A	3
2	B	1
3	C	2
4	D	

Employees Details

EMP E2

E2.EId	E2.Ename	E2.MGR
1	A	3
2	B	1
3	C	2
4	D	

Manager Details

Employees Details			Manager Details		
E1.EId	E1.Ename	E1.MGR	E2.EId	E1.Ename	E1.MGR
1	A	3	3	C	2
2	B	1	1	A	3
3	C	2	2	B	1
4	D				

E1.MGR = E2.EId

realme Shot by ❤SANDIP❤

Syntax:

- Oracle: Select Column_Name from TN_1, TN_2 where <Join_Condition>;

  $E_1.MGR = E_2.EId$

- ANSI: Select Col_name from TN_1 Join TN_2 on <Join_Condition>;

(2) WAQTD employees salary and their manager salary.

Select E_1.Sal, E_2.Sal from emp E_1, emp E_2 where E_1.MGR = E_2.~~Eid~~ empno;

(3) WAQTD employees name, emps salary, manager name and manager salary.

Select E_1.Ename, E_2.Sal, ~~from emp~~ E_2.Ename, E_2.Sal from emp E_1, emp E_2 where E_1.MGR = E_2.empno;

(4) WAQTD employees name, manager name if emps are working as clerk and manager's are working as Analyst.

Select E_1.Ename, E_2.Ename from emp E_1, emp E_2 where E_1.MGR = E_2.empno And E_1.job = 'CLERK' And E_2.job = 'ANALYST';

(5) WAQTD emps name, emps sal, manager name, manager sal if emps are getting sal less than manager sal.

realme Shot by ❤️SANDIP❤️
realme 8i 2024.02.11 12:39

Select E1.Ename, E2.Ename, E1.sal, E2.sal from emp E1, emp E2 where E1.mgr = E2.empno and E1.sal < E2.sal;

(6) WAQTD emps name, emps hiredate, manager name, manager's hiredate if emps are hired before manager.

Select E1.Ename, E1.~~sal~~ hiredate, E2.Ename, E2.~~sal~~ hiredate from emp E1, emp E2 where E1.mgr = E2.empno and E1.hiredate < E2.hiredate;

(7) WAQTD emps name, manager name and manager's manager name.

Select E1.Ename, E2.ename, E3.ename from emp e1, emp e2, emp e3 where e1.mgr = e2.empno and ~~mgr In (select empno from emp where mgr In (select empno from emp))~~; and e2.mgr = e3.empno;

(8) WAQTD emps name, and their deptname

~~Select ename from emp where deptno In (select~~

Select ename, dname from emp, dept where emp.deptno = dept.deptno;

(9) WAQTD emps name, emps deptname, manager name, manager's deptname.

Select e1.ename, d1.dname, e2.ename, d2.dname from emp e1, dept d1, emp e2, dept d2 where e1.deptno = d1.deptno And e2.deptno = d2.deptno And e1.mgr = e2.empno;

(10) WAQTD emps name, emps deptname, mgr name, ~~mgr dept name~~, manager manager's name and manager manager's deptname.

select e1.ename, e2.ename, e3.ename, d1.dname, d2.dname, d3.dname from emp e1, emp e2, emp e3, dept d1, dept d2, dept d3
where e1.deptno = d1.deptno and e2.deptno = d2.deptno and e3.deptno = d3.deptno and e1.mgr = e2.empno and e2.mgr = e3.empno;

Joining of Multiple Tables:

(N-1)

★ ① emp ←(emp.deptno = dept.deptno)→ dept

★ ② emp e1 ←(e1.mgr = e2.empno)→ emp e2

★ ③ emp e1 ←(e1.mgr = e2.empno)→ emp e2 ←(e2.mgr = e3.empno)→ emp e3

★ ④ emp e1 ←(e1.deptno = d1.deptno)→ dept d1
emp e2 ←(e2.deptno = d2.deptno)→ dept d2
e1.mgr = e2.empno

★ ⑤ emp e1 ←(e1.deptno = d1.deptno)→ dept d1
emp e2 ←(e2.deptno = d2.deptno)→ dept d2
emp e3 ←(e3.deptno = d3.deptno)→ dept d3
e1.MGR = e2.empno
e2.mgr = e3.empno

Home Work

① WABTD emps name, emps sal, emps hiredate, emps jobs, emps deptname, manager name, manager sal, manager hiredate, manager job, manager loc, manager manager's name, manager manager's sal, mgr mgr's job, mgr mgr's hiredate, mgr mgr's deptname and emps are working in the Sales dept ~~emps~~ manager's are working in Sales dept and manager's mgr are working in accounting and emps are getting

Sal less than mgr and emps are working as Salesman and emps are getting comm and emps name starts with A, M, J and their deptname starts with S and ends with S.

Assignment on Self Join

① WAQTD name of the employee and his manager's name if employees is working as clerk.

Select E1.Ename, E2.Ename from Emp E1, Emp E2 where E1.MGR = E2.Empno and E1.job='CLERK';

② WAQTD name of the employee and manager's designation if manager works in dept 10 OR 20.

Select E1.Ename, E2.job from Emp E1, Emp E2 where E1.Mgr = E2.Empno and (E2.deptno = 10 OR E2.deptno = 20);

③ WAQTD name of the emp and managers salary if employee and manager both earn more than 2300.

Select E1.Ename, E2.Sal from Emp E1, Emp E2 where E1.Mgr = E2.Empno and (E1.Sal > 2300 and E2.Sal > 2300);

④ WAQTD emp name and manager's hiredate if employee was hired before 1982.

Select E1.Ename, E2.hiredate from Emp E1, Emp E2 where E1.Mgr = E2.Empno and E1.hiredate < '01-JAN-1982';

⑤ WAQTD emp name and manager's comm if employee works as salesman and manager works in dept 30.

Select E1.Ename, E2.comm from Emp E1, Emp E2 where E1.Mgr = E2.Empno and (E1.job = 'SALESMAN' and E2.deptno = 30);

realme Shot by ❤SANDIP❤

⑥ WAQTD emp name and manager name and their salaries if employee earns more than manager.

Select E1.Ename, E1.Sal, E2.Ename, E2.Sal from Emp E1, Emp E2 where E1.Mgrn = E2.Empno and E1.Sal > E2.Sal;

⑦ WAQTD emp name and hiredate, manager name and hiredate if manager was hired before employee.

Select E1.Ename, E1.hiredate, E2.Ename, E2.hiredate from Emp E1, Emp E2 where E1.Mgrn = E2.Empno and E2.hiredate < E1.hiredate;

⑧ WAQTD emp name and manager name if both are working in same job.

Select E1.Ename, E2.Ename from Emp E1, Emp E2 where E1.Mgrn = E2.Empno and E2.job = E1.job;

⑨ WAQTD emp name and manager name if manager is working as Actual manager.

Select E1.Ename, E2.Ename from Emp E1, Emp E2 where E1.Mgrn = E2.Empno and E2.job = 'MANAGER';

⑩ WAQTD emp name and manager name along with their Annual salaries if employee works in dept 10, 20 and manager's sal is greater than employees salary.

Select E1.Ename, E1.Sal*12, E2.Ename, E2.Sal*12 from Emp E1, Emp E2 where E1.Mgrn = E2.Empno and E1.dept in (10, 20) and E2.Sal > E1.Sal;

realme
Shot by ❤ SANDIP ❤

(11) WAQTD emps name and manager's designation for all the employees.

Select E1.Ename, E2.job from Emp E1, Emp E2 where E1.Mgr = E2.Empno;

(12) WAQTD employees name and manager's salary for all the employees if manager's salary ends with 50.

Select E1.Ename, E2.Sal from Emp E1, Emp E2 where E1.Mgr = E2.Empno and E2.Sal like '%50';

Homework

Select E1.Ename, E1.~~Ename~~Sal, E1.hiredate, E1.job, D1.Dname, E2.Ename, E2.Sal, E2.hiredate, E2.job, D2.Loc, E3.Ename, E3.Sal, E3.hire-date, D3.Dname from Emp E1, Emp E2, Emp E3, Dept D1, Dept D2, Dept D3 where E1.Mgr = E2.Empno and E2.Mgr = E3.Empno and E1.deptno = D1.deptno and E2.deptno = d2.deptno and E3.deptno = d3.deptno and d2.dname = 'SALES' and d1.dname = 'SALES' and d3.dname = 'ACCOUNTING' and e1.sal < e2.sal and e1.job = 'SALESMAN' and e1.comm is not null and (e1.ename like 'A%' or e1.ename like 'M%' or e1.ename like 'J%') and d1.dname like 'S%' and d1.dname like '%S';

realme Shot by ❤SANDIP❤
realme 8i 2024 02 11 12:39

21.08.23

For Emp's

Ename	SAL	Hiredate	Job	DName
ALLEN	1600	20-FEB-81	Salesman	Sales
MARTIN	1250	01-MAY-81	Salesman	Sales

For Manager

~~EName~~	Sal	Hiredate	Job	Loc
BLAKE	2850	28-SEP-81	Manager	Chicago
BLAKE	2850	28-SEP-81	Manager	Chicago

For Manager's manager

EName	Sal	Hiredate	Dname
King	5000	17-Dec-81	Accounting
King	5000	17-Dec-81	Accounting

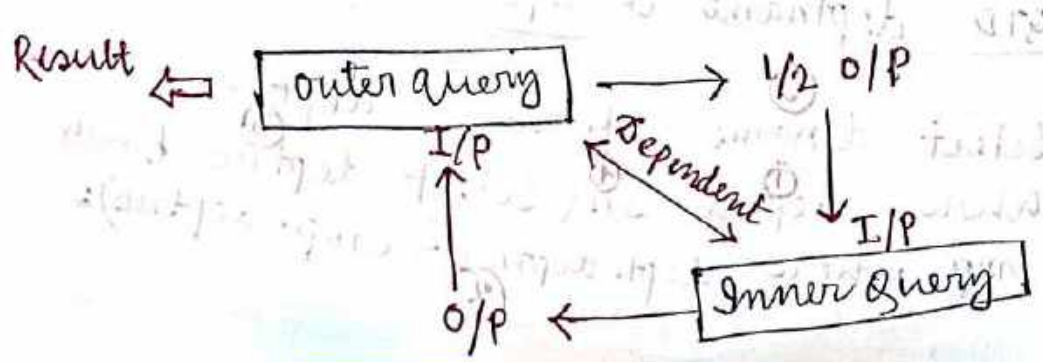
① WAQTD deptname of emps

→ Select dname from dept where deptno In (Select deptno from emp);

Combination of joins and subquery

→ Select dname from dept, emp where emp.deptno = dept.deptno;

☒ Co-Related Sub-Query: ① A Query written inside another query ~~Search~~ Such that outer query and inner query are dependent on each other or (Interdependent on each other) is known as Co-Related Sub-query.



realme Shot by ❤SANDIP❤

Working process of Co-Related Subquery

① Let us, consider two queries - outer query and Inner query.

② First, outer query will execute and produces the partial output or half output.

③ The partial output of outer query is passed as input to the inner query by taking that i/p inner query will produces the output.

④ The o/p of inner query is passed as input to the outer query.

⑤ By taking that input outer query will produce the result.

Therefore we can state that, outer query and inner query both are interdependent on each other.

Note: ① Co-Related Subquery works with the principle of both joins as well as subquery.

② In co-related subquery joins condition is mandatory and join condition has to be written inside inner query.

① WAQTD deptname of emps

```
Select dname from dept
where deptno in (Select deptno from
emp where dept.deptno = emp.deptno);
```

realme Shot by SANDIP

Dept

Dname	Dno
D1	10
D2	20
D3	30
D4	40

Emp

Ename	Dno
A	10
B	20
C	30
D	20

Executing Process

Step-1:

10

20

30

40

Step-2:

10 = 10

20 = 20

30 = 30

40 = 20

i) $\boxed{10 = 10}$
10 = 20
10 = 30
10 = 20

ii) 20 = 10
$\boxed{20 = 20}$
20 = 30
$\boxed{20 = 20}$

iii) 30 = 10
30 = 20
$\boxed{30 = 30}$
30 = 20

iv) 40 = 10
40 = 20
40 = 30
40 = 20

Null

Step-3:

10

10,20

30

Null

Step-4:

IN

$\boxed{10 = 10}$

$\boxed{20 = 20, 20}$

$\boxed{30 = 30}$

40 = Null

Step-5:

D1
D2
D3

VI Sub-Query operator:

① **Exists:** ① Exists operator is a unary operand (one operand) which can accept one operand towards RHS and that one operand has to be a Co-Related Subquery.

② Exists operator returns true if the subquery returns any value other than null value.

① WAQTD departname which emps are workings

```
Select dname from dept where exists (
Select * from emp where emp.deptno =
dept.deptno);
```

realme Study SANDIP
realme 8i 2024.02.11 12:40

② **Not Exists:** ① Not Exists operator is a unary operand (one operand) which can accept one operand towards RHS and that one operand has to be a co-related subquery.

② Not Exists operator returns true if the subquery returns only null values.

① VAQTD dname of emps in which emps are not working.

Select dname from dept where Not Exists (Select deptno from emp where emp.deptno = dept.deptno);

IV N^{th} Max and N^{th} Min:

> N^{th} Max:

Syntax: Select sal from ~~E1.sal~~ / Emp E_1 where N-1 = (Select count (distinct sal) from ~~E2.sal~~ / Emp E_2 where (E_1.sal < E_2.sal));

> N^{th} Min:

Syntax: Select sal from Emp E_1 / ~~E1.sal~~ where N-1 = (Select count (distinct sal) from Emp E_2 / ~~E2.sal~~ where (E_1.sal > E_2.sal));

Example: Select sal from Emp E_1 where (Select count (distinct sal) from Emp E_2 where E_1.sal < E_2.sal) In (0, 1, 3, 7);

realme Shot by ❤ SANDIP ❤

Single-row function:

1. Length()
2. Concat()
3. Upper()
4. Lower()
5. Initcap()
6. Reverse()
7. Substr()
8. Replace()
9. Instr()
10. Round()
11. Trunc()
12. Mod()
13. Last_Day()
14. Months_Between()
15. To_char()
16. NVL() [Null value Logic]

Length(): It is used to find the number of characters present in the given string.

> Syntax:

```
LENGTH ('string')
```

① WAQTD no. of characters present in MARTIN.

```
Select length (ename) from emp
where ename = 'MARTIN';
```

② WAQTD length of names of emps.

```
Select length (ename) from emp;
```

③ WAQTD length of names of emps along emps names.

```
Select length (ename), ename from emp;
```

④ **DUAL:** It is a dummy table which consists of one row and one column

① WAQTD length of name of Sandip.

```
Select length ('SANDIP') from dual;
```

Concat(): It is used to join two strings

> Syntax: CONCAT ('string1', 'string2')

realme
SHOT ON ❤️ SANDIP ❤️

① WAQTD join two name Sandip and Mandal

Select concat ('Sandip', 'Mandal') from dual;

Upper(): It is used to convert the given string into uppercase.

> Syntax: Upper ('string')

① WAQTD convert Sandip name as uppercase

Select upper ('sandip') from dual;

Lower(): It is used to convert the given string into lowercase.

> Syntax: Lower ('string')

① WAQTD convert SANDIP name as lowercase.

Select Lower ('SANDIP') from dual;

WAQTD convert ename as lowercase.

Select Lower (ename) from emp;

Initcap(): It is used to convert the given string first character into uppercase.

Syntax: Initcap ('string')

① WAQTD convert the ename column first character to uppercase.

Select Initcap (ename) from emp;

Reverse(): It is used to reverse the given string.

Syntax: Reverse ('string')

realme Shot by ❤SANDIP❤

① WAQTD reverse name of Sandip.
Select reverse ('Sandip') from dual;

■ substr(): It is used to extract a part of string from the given original string.

> Syntax: Substr ('original_string', position [, length])

Note: ① length is optional if the length is not mentioned it will consider the entire string.

-10 -9 -8 -7 -6 -5 -4 -3 -2 -1
STRAWBERRY
1 2 3 4 5 6 7 8 9 10

i) ('Strawberry', 6, 4) ⇒ BERR
ii) ('Strawberry', 1, 5) ⇒ STRAW
iii) ('Strawberry', -5, 5) ⇒ BERRY

① Select substr ('Strawberry', 6, 4) from dual;
→ BERR

*② WAQTD first two character of emps name;
Select substr (ename, 1, 2) from emp;

*③ WAQTD last two character of emps name
Select substr (ename, ~~1, 2~~ -2) from emp;

*④ WAQTD first half of emps name
select substr (ename, 1, length (ename)/2) from emp;

■ Replace(): It is used to replace a string from the given original string.

Syntax:
Replace ('original_string', 'string' [, 'New_string'])

realme Shot by SANDIP
realme 8i 2024 02 11 12:40

Note: If the third argument is not mentioned it will considered the null value.

i) Replace ('SANDIP', 'S', 'M') ⇒ MANDIP
ii) Replace ('SANDIP', 'S', 'ABC') ⇒ ABCANDIP
iii) Replace ('SANDIP', 'SA', '123') ⇒ 123NDIP

Round(): It is used to round of the given number based on the scale value.

Syntax: Round(Number[, scale])

① Select round(3.4) from dual;

Round(3.4)
3

② Select round(3.5) from dual;

Round(3.5)
4

③ Select round(33.5, 9) from dual;

Round(33.5,9)
33.5

Trunc(): It is similar to round() function instead of going to the highest value it will round to the lower value.

Syntax: Trunc(Number[, scale])

① Select trunc(4.5) from dual;

Trunc(4.5)
4

② Select trunc(4.9) from dual;

trunc(
4

realme Shot by ❤SANDIP❤
realme 8i 2024.02.11 12:40

Mod(): It is used to obtained the modulus or reminder of a given number.

Syntax: mod(m, n)

① Select mod (2,2) from dual;

mod (2,2)

0

② Select mod (5,2) from dual;

mod (5,2)

1

> Note: • sysdate: It is used to obtained today's date.

• current-date: It is used to obtained ~~today's~~ the current system date.

• Systimestamp: It is used to obtained date, time and time zone.

Months_Between(): It is used to obtained the number of months present between two dates.

Syntax: Months_Between ('date', 'date')

① Select months_between ('23-Aug-23', '23-Aug-24') from dual;

months_between ('23-Aug-23', '23-Aug-24')

-12

Last-day(): It is used to obtained the last day of the given particular date.

Syntax: Last_day ('date')

① Select last_day (current_date) from dual;

last_day (

31-AUG-2023

realme Shot by ❤SANDIP❤
realme 8i 2024 02 11 12:40

① **To_char():** It is used to Obtained the given date into a String format based on the model given.

> **Format model:**

year: Twenty Twenty Three
YYYY: 2023
YY: 23
MONTH: August MM: 08
MON: Aug
DAY: Wednesday
DY: Wed
D: 4
HH24: 11 hours
HH12: 11 hours
MI: 16 minutes
SS: 45 Seconds
HH12: MI: SS: 11:16:45
HH24: MI: SS: 11:16:45
'MM-DD-YYYY': 08-23-2023
DD-MM-YY: 23-08-23

Syntax: To_char(date, 'Format_model')

② **NVL():** It is used to avoid the null values or Side Effects of null values in the arithmetic operations.

Syntax: NVL (Arg1, Arg2);

> **Argument 1:** Here, we write the column name or expression which can result in null.

> **Argument 2:** i) Here, we write a numerical value which can be substitute with Arg1 if the Arg1 is null value.

ii) If the argument1 is not a null value then the same value will be considered.

realme Shot by ❤SANDIP❤

NVL(Argument 1, Argument 2)

Select [Sal+NVL(Comm,0)] ~~from emp~~;

Sal	Comm
100	10
200	
300	20
400	

NVL0
NVL0
NVL0

Result Table

Sal+Comm
110
200
320
400

Instr(): ① It is used to obtained the position of a string in the Original String.

② It is used to search for a string in the original string if the string is present it returns true else, it returns zero.

Syntax: Instr('Original string', 'string', Position [, Occurance])

① **Note:** If Occurance is not mentioned the default value will be 1.

① Select Instr('MALAYALAM', 'LA', 1,1) from dual;

1	2	3	4	5	6	7	8	9
M	A	L	A	Y	A	L	A	M

Instr

3

Pseudo Column: It is the false column which is present in each and every table and must be called explicitly

We have two pseudo columns

① RowNum
② RowId

RowNum: ① RowNum is the serial number that is assigned to the result table.

② RowNum is assigned @ time of executions.

realme 6i 2024 02 11 12:40

③ RowNum is always starts with 1.
④ RowNum is dynamic (Not constant)

① WAQTD rownum, name, sal of emps who are working as manager.

Select rownum, name, sal from emp where job = 'MANAGER';

Rowid: ① Rowid is the 18 digits address of the record in the memory.

② Row Id is generated @ time of Insertion of the record.

③ Row Id's are unique.

④ Row Id's won't be deleted.

⑤ Row Id's are static (constant).

⑥ Row Id's the easiest way to access or delete a record.

① WAQTD rowid of JONES.

Select rowid from emp where Ename = 'JONES';

② WAQTD first three records of emps.

Select * from emp where rownum In (1,2,3);

Select * from emp where rownum < 4;

③ WAQTD 3rd record of emp;

Select * from ~~emp~~ ~~where~~ ~~rownum~~ (Select rownum as "SNO", emp.* from emp) where ~~rownum~~ SNO = 3;

④ WAQTD last record of emp table.

→ Select * from (Select rownum as "SNO", emp.* from emp) where ~~rownum~~ SNO = 14;

Another method → Select * from (Select emp.*, rownum as "SNO" from emp) where SNO In (Select Count(*) from emp);

realme
Photo by ❤️SANDIP❤️

(5) WAQTD last but first record of emps.

Select * from (Select emp.*, rownum as "SNO" from emp) where SNO In (Select count(*)-1 from emp);

(6) WAQTD first 5 records.

Select * from emp where rownum In (1,2,3,4,5);
Select * from emp where rownum < 6;

(7) WAQTD odd no of emps records.

Select * from ~~emp~~ (Select emp.*, rownum as "SNO" from emp) where mod (SNO, 2) = 1;

(8) WAQTD even no of emps records.

Select * from (Select emp.*, rownum as "SNO" from emp)
where mod (SNO, 2) = 0;

(9) WAQTD last 50% of emps records.
/WAQTD last half of emps records

Select * from (Select rownum as "SNO", emp.* from emp)
where SNO > (Select count(*)/2 from emp);

(10) WAQTD first 50% of emps records.

Select * from (Select rownum as "SNO", emp.* from)
where SNO <= (Select count(*)/2 from emp);

(11) WAQTD last 50% of records and in emps names contain character 'A' in it.

Select * from (Select rownum as "SNO", emp.* from emp)
where SNO > (Select count(*)/2 from emp)
and ename like '%A%';

realme Shot by ❤SANDIP❤
realme 8i 2024 02 11 12:41

(12) WAQTD first max (sal) of emps.

```
Select sal from emp E1
where sal (select count (distinct sal)
from emp E2 where E1.sal < E2.sal)
In (0);
```

(13) ☆☆ WAQTD 3rd max(sal) of emps.

Based on Pseudo Columns

```
Select sal
from (select rownum as "SNO", sal
from (select distinct sal from emp
order by sal desc))
where SNO = 3;
```

(14) WAQTD details of emps who are getting 2nd max sal.

```
~~Select sal~~ select * from
~~from (select~~ emp where sal In (select
sal from (select rownum as "SNO", sal
from (select distinct sal from emp
order by sal desc))
where SNO = 2);
```

(15) WAQTD details of emps who are getting 2nd max (sal)

(16) WAQTD top 5 min sal

```
Select sal from (select rownum as
"SNO", sal
from (select distinct sal from emp
order by sal desc))
where SNO In (1,2,3,4,5);
```

(17) WAQTD top 3 max sal

```
Select sal from (select rownum as
"SNO", sal
from (select distinct sal from emp
order by sal desc))
where SNO In (1,2,3);
```

(18) WAQTD first, 3rd and 5th min sal of emps.

```
Select sal from (select rownum as 'SNO'
~~from~~, sal
from (select distinct sal from emp
order by sal ASC))
where SNO in (1, 3, 5);
```

(19) *** WAQTD 1st max and last max. sal of emps.

```
Select sal from (Select rownum as "SNO",
sal from (Select distinct sal from emp
order by sal desc))
where SNO in 1 or SNO (Select
count (distinct sal) from emp);
```

Overview of SQL Statements:

1. DDL (Data Definition Language)
2. DML (Data Manipulation Language)
3. DCL (Data Control Language)
4. TCL (Transaction Control Language)
5. DQL (Data Query Language)

DDL (Data Definition Language): It is used to construct or create an object in the database and it will deals with the structured of the object.

> 5 statements:

1. CREATE
2. ALTER
3. RENAME
4. TRUNCATE
5. DROP

CREATE: It is used to create an object into the database.

① Syntax:

```
Create Table Table_Name
(
  Col1, datatype, Constraint,
  Col2, data type, Constraint,
  :
  Coln, datatype, Constraint
);
```

Example:

```
Create Table Student
(
  SId Number(2), primary Key,
  Sname varchar(20), NOT NULL,
  SPhno Number(10) check(length(SPhno)
                        =10),
  SDOB Date,
  Gen char(2)
);
```

② ALTER: It is used to change the structure of an object/table.

i) To add a column

> Syntax:

```
Alter Table Table_Name
Add Column_Name Datatype
Constraint;
```

Example:

```
Alter Table Student
Add EmailId varchar(20);
```

ii) To Drop a column

> Syntax:

```
Alter Table Table_Name
Drop column Column_Name;
```

Example:

```
Alter Table Student
drop column EmailId;
```

iii) To Rename a column

> Syntax:

```
Alter Table Table_Name
Rename Column Column_Name to
        New Column_Name;
```

realme Shot by SANDIP

Example: Alter table Student
Rename Column SId to Student_Id;

iv) To Modify the Data Types:

> Syntax: Alter Table Table_Name
Modify Column_Name New
Datatype;

Example: Alter table Student
Modify gen varchar(2);

v) To Modify the Constraints:

> Syntax: Alter Table Table_Name
Modify Column_Name existing_Data
Type Null/Not Null;

Example: Alter table Student
Modify Sdob date not null unique;

Rename: It is used to rename the table name.

> Syntax: Rename Table_Name
To New Table_Name;

Example: Rename Student to Employee;

Truncate: It is used to delete all the records from the table permanantly.

> Syntax: Truncate Table Table_name;

Drop: It is used to delete the table but the table is present in the recycle bin.

> Syntax: Drop Table Table_name;

FLASH BACK: It is used to get back the table from the recycle bin.

> Syntax: FlashBack Table Table_Name
to Before Drop;

PURGE: It is used to delete the table from the recycle bin.

> Syntax: `Purge Table Table_Name;`

Note: DDL is a autocommit statement.

DML (Data Manipulation Language):

It is used to manipulate the object by performing insertion updation and deletion. It has three statements

1. INSERT
2. UPDATE
3. DELETE

① **INSERT:** It is used to insert the records into the table

> Syntax:

```
INSERT into Table_Name values
        (v1, v2, v3 ... vn);
```

Example:

```
Insert into Employee
values(101, 'SANDIP', null, 'M', '21-DEC-2001');
```

> Syntax 2:

```
Insert Into Table_Name Values
        (&Col1, &Col2, &Col3, &Col4 ... &Coln);
```

```
Enter value for coln1:-
Enter value for coln2:-
:
Enter value for colns:-
```

Example:

```
Insert Into Employee Values
        (&EId, &Ename, &EPhono, &EGen);
Enter value for EId: 101
Enter value for Ename: 'SANDIP'
Enter value for EPhono: 6295510306
Enter value for EGen: 'M'
```

UPDATE: It is used to modify and existing value.

> Syntax:

```
Update Table_Name
  Set Column_Name1 = Values
  , Column_Name2 = Values
  [where = statement];
```

Example: update Employee
Set EId = 102, EGen = 'M'
where EId = 101;

DELETE: Delete is used to delete a particular record from the table.

> Syntax: Delete ~~Table~~ Table_Name
From
[where = statement];

Example: ① Delete from Employee
where EId = 101;
② Delete from Employee;

Note: DML is not a auto commit statement.

TCL (Transaction Control Language): ① It is used to Control the transaction on the database.

② The transaction done on the database are known as insertion, updation and deletion.

③ It has three statements
i) COMMIT
ii) SAVEPOINT
iii) ROLLBACK

COMMIT: It is used to save the transaction into the database.

> Syntax: COMMIT

Example: Insert into Employee
Values (103, 'DHONI', null, null, '07-JUL-1981');
1 row created
SQL> commit
Commit complete

- SAVEPOINT: This statement is used to mark the position or restoration points (nothing related to database)

> Syntax: SAVEPOINT Savepoint_name;

Example: Insert into Employee
Values (104, 'KHOLI', null, null, '08-JUL-1980');
1 row created
SQL> Savepoint S1
Savepoint created

- ROLLBACK: ① It is used to obtained the save data from the database.
② It will bring back you to the point where you have committed for last time.

> Syntax: ROLLBACK

It is used to take back you to the ~~same~~ save point name.

> Syntax2: ROLLBACK To SavePoint name;

DCL (Data Control Language): ① It is used to control the data flow between the users.

② It has two statements
① GRANT.
② REVOKE.

① **GRANT:** It is used to give the permission or grant the permission to the user.

> **Syntax:** GRANT Select Statement
> ON Table_Name
> TO User_Name;

```
SQL> connect
Enter user-name: SCOTT
Enter Password: tiger
Connected
SQL> Show user;
USER is "SCOTT"
SQL> grant select on emp to hr;
Grant Succeeded
```

```
SQL> connect
Enter user-name: hr
Enter password: tiger
Connected
SQL> show user;
USER is "hr"
SQL> Select * from scott.emp;
```

② **REVOKE:** It is used to get back the permission from the user.

> **Syntax:** Revoke Select Statement
> ON Table_Name
> From User_Name;

SQL> show user
User is "SCOTT"

SQL> revoke select on emp from hr;
Revoke succeeded.

To copy a Table:

Syntax: Copy table table_name
As
Select statement;

To create a view:

① View is the virtual table which will be created on the result of a query.

② It doesn't occupy any space.

③ It can be created on any table (one or more).

④ Any changes done on the original table will have an impact on the view table.

Syntax: Create view viewname
As
Select statement;

SQL> connect
Enter user-name: System
Enter password: tiger
Connected

SQL> grant create view to scott;
Grant succeeded

SQL> connect
Enter user name: SCOTT
Enter password: tiger
connected

SQL> create view emp_v
as select * from emp where sal > 3000;
view created

realme Shot by SANDIP

To copy a Table:

Syntax:
```
Create Table Table_name
as
Select Statement;
```

Example:
```
Create table analyst
as
Select * from emp
where job = 'ANALYST';
```

table created

Set operators:

① It is used to combine the two select statements and gives the result.

② It has 4 types
- i) UNION
- ii) UNION ALL
- iii) INTERSECT
- iv) MINUS

i) **union:** i) It is used to combine the result of two statements and it will remove the duplicated values.

ii) From both the queries the column name and data type has to be same.

Syntax:
```
Select Column_Name
From Table_Name
UNION
Select column_Name
From Table_Name;
```

Example:
```
Select deptno from ~~dep~~emp
union
Select deptno from dept;
```

ii) **union all:** It is used to combine the two select statements result and values. it will not remove the duplicated

Syntax:
```
Select column_name
From Table_name
UNION ALL
Select column_Name
from Table_Name;
```

Example: Select deptno from emp
union all
Select deptno from dept;

iii) Intersect: It is used to combine the result of two select statements and it will give the common records between the two statements.

> Syntax:

```
Select Column name
from Table name
Intersect
Select Column Name
from Table name;
```

Example: Select deptno from emp
Intersect
Select deptno from dept;

iv) Minus: It is used to combine the result of two select statements and it will give the records present in the table 1 and the record which is not present in the table 2.

> Syntax:

```
Select Column name
from Table name
Minus
Select Column Name
from Table name;
```

Example: Select deptno from emp
Minus
Select deptno from dept;

Attributes: ① This are the properties which describes an entity.

② There are 7 types of attributes:

i) Key / candidate attributes
ii) Non-Key attributes
iii) ~~Prim any~~ Prime Key attributes
iv) Non-Prime Key attributes
v) Composite Key attributes

vi) Super key attributes
vii) Foreign key attributes

① **Key attribute/candidate key:** A attribute which is used to identify a record uniquely from a table is known as key attribute.

Ex: phone_No, mail-id, aadhar, pan, ration, passport, dl, bank a/c.

② **Non key attribute:** All the attributes other than key attributes.

Ex: Name, age, gender, dob.

③ **Prime key attribute:** Among the key attributes an attribute is chosen to be the main attribute to identify a record uniquely from the table is known as prime key attribute.

Ex: Phone_No.

④ **Non-prime key attribute:** All the key attributes other than prime key attributes.

Ex: maild-id, addhar, pan, ration, passport, dl, bank a/c.

⑤ **Composite key attribute:** It is combination of two or more ~~than~~ non key attributes, which is used to identify a record uniquely from the table.

> (composite key attribute) is found whenever there is no key

Ex: (name, age, dob, address)

⑥ **Super key attribute:** It is a set of all key attributes.

Ex: {phone_No, mail-id, aadhar, pan, ration, passport, dl, bank a/c}

⑦ **Foreign key attribute:** It is an attributes which behaves as an attribute of another entity to represent the relationship.

Ex: Dno

Attributes

Name, age, color, weight, height, occupation, Nationality, address, gender, dob, job, mail_id, aadhar, phone no, pan, ration, dl, bank a/c...

Human Being

Entity

Functional Dependency:

"There exists a dependency such that an attribute in a Relation determines another attribute."

Example:

EMP-(EID, ENAME)

EID --> ENAME : functional dependency

Types of Functional Dependencies:

1. Total Functional Dependency
2. Partial Functional Dependency
3. Transitive Functional Dependency

1. Total Functional Dependency:

If an attribute in a relation determines all the other attributes it is known as TFD

OR

If all the attributes are dependent on a single attribute then it is known as TFD.

EMP-(EID, ENAME, SAL, DOB)

EID* KEY ATTRIBUTE

EID -> ENAME

EID -> SAL

EID -> DOB

EID --> (ENAME, SAL, DOB) :- total functional dependency.

② Partial Functional Dependency:

There ~~are~~ exists a dependency such that a part of composite key attributes determines another attribute uniquely.

CUSTOMER-(CNAME, ADDRESS, MAIL-ID, PHONE-NO)

CUSTOMER

CNAME	ADDRESS	MAIL-ID	PHONE-NO
Smith	Mysore	smith@gmail.com	
Miller	Bangalore		1001
Scott	Mangalore	scott@yahoo.com	
Adams	Mysore		2002
Scott	Delhi	scott@yahoo.com	3003

(Phone-No, Mail-id) --- Composite Key attribute

PHONE-NO --> CNAME, ADDRESS
MAIL-ID --> CNAME, ADDRESS :- partial functional dep.

③ Transitive Functional Dependency:

There ~~are~~ exists a dependency such that an attribute is determined by a non-key attribute, which is intern determined by a key attribute.

CUSTOMER-(CID, CNAME, PINCODE, CITY)

CUSTOMER

CID	CNAME	PINCODE	CITY
1	Smith	560019	Bangalore
2	Miller	560019	Bangalore
3	Scott	312121	Pune
4	Adams	123456	Mumbai
5	Scott	123456	Mumbai

Key attribute

Non Key attribute

CID - CNAME
- PINCODE
- CITY

Redundancy: The repetition of unwanted data is known as redundancy.

Anomaly: The side effects caused during DML operations is known as Anomaly.

TOTAL	PARTIAL	TRANSITIVE
Non Redundancy	Redundancy Exists	Redundancy Exists.
No Anomalies	Anomalies are Present	Anomalies are present

Normalization

> **What is Nomalization?**

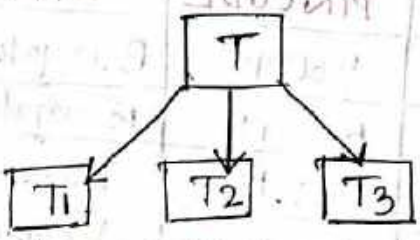
: It is the Process of reducing a large table into smaller tables in order to remove redundancies and anomalies by identifying their functional dependencies is known as Nomalization."

OR

: The process of decomposing a large table into smaller table is known as Normalization."

OR

: Reducing a table to its Normal Form is known as Normalization."



> **What is Normal Form?**

A table without redundancies and anomalies are said to be in Normal Form.

Levels of Normal Form:

1. First Normal Form (1NF)
2. Second Normal Form (2NF)
3. Third Normal Form (3NF)

④ Boyce - Codd Normal Form (BCNF)

Note: If any table/entity is reduced to 3NF, then the table is said to be normalized.

① First Normal Form (1NF):

- No duplicates records.
- Multivalued data should not be present.

QSPIDERS

QID	NAME	COURSE
1	A	JAVA
2	B	JAVA, SQL
3	C	MT, SQL
1	A	MT

→

QID	NAME	C1	C2	C3
1	A	JAVA		MT
2	B	JAVA	SQL	
3	C		SQL	MT

② Second Normal Form (2NF):

- Table should be in 1NF
- Table should not have partial Functional Dependency

EMPLOYEE - (EID, ENAME, SAL, DEPTNO, DNAME, LOC)

Eid	Ename	Sal	DeptNo	DName	Loc
1	A	100	10	D1	L1
2	B	120	20	D2	L2
3	C	320	10	D1	L1
4	D	251	10	D1	L1

Eid - ename, Sal

Deptno - dname, Loc

(Eid, deptno) → (Ename, sal, Dname, Loc)

composite Key attribute results in PFD.

R1 - (EID, ENAME, SAL)

R2 - (DEPTNO, DNAME, LOC)

Eid	Ename	Sal	DNo
1	A	100	10
2	B	120	20
3	C	320	10
4	D	251	10

Deptno	dname	Loc
10	D1	L1
20	D2	L2

3. Third Normal Form (3NF):

- Table should be in 2NF.
- Table should not have Transitive Functional Dependency.

Employee - (EID, Ename, Sal, Comm, Pincode, State, Country)

EID → ENAME :- Transitive Functional Dep
SAL
COMM
PINCODE → STATE
COUNTRY

R1 - (eid, ename, comm)
R2 - (Pincode, State, Country)

First Normal Form (1NF) —1NF→ Second Normal Form (2NF) —2NF→ Third Normal Form (3NF)

1NF
- Single atomic value in each column
- Each row have unique identifier

2NF
- Satisfy all 1NF conditions
- partial dependencies must be removed from the table.

3NF
- Satisfy all conditions of 2NF
- Transitive dependency of non-key attributes or key column must be removed.

PSEUDO COLUMNS ASSIGNMENT:

① WAQTD 5th record from the employee Table.

Select * from (select rownum sno, emp.* from emp) where sno = 5;

② WAQTD Last record from the employee table and having character 'E' in their name.

Select * from (select rownum sno, emp.* from emp) where sno = (select count(*) from emp) and ename like '%E%';

③ WAQTD 3, 4, 5 record from the employee table.

Select * from (select rownum sno, emp.* from emp) where sno In (3, 4, 5);

④ WAQTD 6th maximum salary.

Select sal from (select rownum sno, sal from (select distinct sal from emp order by desc) where sno = 6;

⑤ WAQTD TOP 1, 3, 5 maximum salaries of employees.

Select sal from (select rownum sno, sal from (select distinct sal from emp order by desc)) where sno In (1, 3, 5);

⑥ WAQTD details of last 3 employees records.

Select * from (select rownum sno, emp.* from emp where sno > (select count(*) - 3 from emp);

⑦ WAQTD name, job, salary of employees earning top 5 maximum salaries.

Select ename, job, sal from (select rownum sno, sal, ename, job from (select distinct sal from emp where order by desc)) where sno ≤ 5;

⑧ WAQTD top 3 minimum salaries of employees.

Select sal from (select rownum sno, sal from (select distinct sal from emp order by sal) where sno ≤ 3;

⑨ WAQTD last 50% of record from employee table.

Select * from (select rownum sno, emp.* from emp) where sno ≥ (select count(*)/2 from emp);

⑩ WAQTD last but one record from employee table.

Select * from (select rownum sno, emp.* from emp) where sno = (select count(*)-1 from emp);

Note: In SQL, DDL (Data Definition Language) statements such as [create], [Alter], [Rename], [Truncate], [Drop] are considered auto-commit statements. This means that, when these statements are executed, they are automatically committed to the database, and there is no need for explicit [Commit] command.

It's important to note that, this behavior is specific to DDL statements. DML (Data Manipulation Language) statement such as [Insert], [update] and [delete] do not auto-commit in PL/SQL.

realme Shot by ❤SANDIP❤

Note: In SQL, DML (Data Manipulation Language) Statement such as [Insert], [update] and [delete] are not auto-commit statements. This means that when these statements are executed, they are not automatically committed to the database. Instead, an explicit [commit] command is required to save the changes made by these statements.

This behavior allows for greater control over transactions and the ability to roll back changes if necessary. It's important to note that this behavior may vary depending on the specific database management system and its configuration.

Special operators:

① WASTD details of emps who are working as clerk, manager, analyst in deptno no 20.

```
Select * from emp
where job In ('CLERK', 'MANAGER', 'ANALYST')
and deptno = 20;
```

② WASTD emps names, sal of the emps who are hired in 80 to 83.

```
Select ename, sal from emp where
hiredate between '01-JAN-80' and '31-dec-83';
```

③ WASTD comm of emps who are getting sal and not having reporting manager and having comm.

```
Select comm from emp where sal is
not null and mgr is not null
and comm is not null;
```

④ WASTD emps names who are not working as actual manager in the deptno 10, 40.

```
Select ename from emp where job not In
'MANAGER' and deptno In (10, 40);
```

⑤ WASTD of the emps with their two months salary whose name ends with 'ER' and having 6 characters in their names.

realme
2024 02 11 12:44

Select ename, sal*2 from emp
where ename like '%ER'
and ename Like '%------%';

⑥ WAQTD name and job of the emps gives alias names as your own name for the ename column who hired after 81 and sal in the range of 2345 to 4352.

Select ename as "SANDIP", job
from emp
where hiredate > '31-dec-81'
and sal between 2345 and 4352;

⑦ WAQTD details of emps with their reporting manager who are not hired between 82 to 85.

Select emp.*, mgr
from emp
where hiredate not between
'01-JAN-82' and '31-dec-85';

⑧ WAQTD all the emps sal with deduction of 45% and has character 'B' in their name and not getting comission.

Select sal, sal - sal*45/100
from emp where ename Like '%B%'
and comm is null;

⑨ WAQTD name of the emps who are hired in 20th date and working in department in 40.

use like operator

Select ename from emp
where extract (day from hiredate)
=20 and deptno =40;

⑩ WAQTD job of the emps whos job starts with 'A' and not getting sal and having comm.

Select job from emp
where job Like 'A%'
and sal is null and
comm is not null;

realme Shot by ❤SANDIP❤

(11) WAQTD location of emps who are working in the department accounting, sales, operations, research in the deptno 40.

```
Select loc from dept
where dname In ('ACCOUNTING', 'SALES', 'OPERA-
TIONS', 'RESEARCH')
and deptno In 40;
```

(12) WAQTD name and deptno of emps who are working in the deptno 20, 30, 40, and they are not working as clerk and name has 5 characters.

```
Select ename, deptno from emp
where deptno In (20, 30, 40)
and job not In ('CLERK')
and ename Like '%-----%';
```

(13) WAQTD emps who are working as salesman in the deptno 10, 20 and display their name and emp id.

```
Select ename, empno
from emp
where job = 'SALESMAN'
and deptno In (10, 20);
```

(14) WAQTD details of emps who are getting comm in between 100 to 5000 and not working as clerk and analyst and manager and salesman.

```
Select * from emp where
comm between 101 and 4999
and comm is not null
and job not In ('CLERK', 'MANAGER', 'ANALYST',
'SALESMAN');
```

(15) WAQTD emply name, deptno, sal of the emps along with their annual sal with deduction of 50% who are working in deptno no 10 and 20 and their annual sal is greater than 35555.

```
Select ename, deptno, sal,
Sal* 12 As "ANNUAL SAL",
Sal*12 - Sal*12* 50/100 "Ded"
where deptno In (10, 20) and Sal*12> 35555;
```

realme
Shot by SANDIP
2024/02/11 12:45

(16) WAQTD deptno of the emps along with their ~~empno who are working~~ mid term salary give alias name as year Salary who's name starts with 'S' And ends with 'H'.

```
Select deptno, Sal*6 as "Salary"
from emp
where ename like 'S%H';
```

(17) WAQTD details of the emps along with their empno who are working in deptno 10, 10, 30 and having the empno in the range of 7369 to 7902.

```
Select emp.*, deptno from emp
where deptno In(10, 40, 30)
and empno between 7369 and 7902;
```

(18) List the emps who are not working in dept 20 and having the mgr in the range of 7698 to 7788.

```
Select * from emp
where deptno not In 20
and mgr is not null
and mgr between 7698 and 7788;
```

(19) WAQTD details of the emps along with their annual Sal who are getting comm in the range between 999 to 1399 and who are not having the reporting manager.

Select emp.*, Sal*12 as "Annual Sal"
from emp where comm is not null and comm
between 1000 and 1398 and mgr is null;

(20) WAQTD empid, comm of the emps who are hired
in 81 and has 'R' letter in their names and
'A' in their job and not working in deptno 10,20,
40.
Select emp.no, comm from emp
where hiredate between '01-JAN-81' and
'31-dec-81' and ename like '%R%'
and job like '%A%' and deptno not In
(10, 20);

(21) List the job of the emps who are not working as
clerk and not working in 10, 20, 30 and
not having reporting manager and having
Commission and not and not having Salary.
Select ename, job from emp where job not
In (10, 20, 30) and mgr is null
and comm is not null
and sal is null;

Subquery:

(1) Display all the employees whose department
names ending 's'.
Select dname from dept
where dname like '%S';

(2) Query to display the employee names who
is having maximum salary in deptname
'Accounting'.

realme Shot by ❤SANDIP❤

select ename from emp where sal In (select max(sal) from emp) and deptno In (select deptno from dept where dname = 'Accounting');

③ Query to display the dept name who is having highest commission.

select dname from dept where deptno In (select deptno from emp where comm In (select max (comm) from emp));

④ Query to display the emps names whose department name has 2nd character as 'O'.

select ename from emp where deptno In (select deptno from dept where dname like '_O%';

⑤ Query to display all the employees who's dept number is same as Scott.

select ename from emp where deptno = (select deptno from emp where ename = 'SCOTT');

⑥ Query to display all the emps in 'Operations and Accounting' dept.

select ename from emp where deptno In (select deptno from dept where dname In ('ACCOUNTING', 'OPERATIONS'));

⑦ List the emps who has salary greater than miller.

select ename from emp where sal> (select sal from emp where ename = 'MILLER

⑧ List department name having atleast 3 Salesman.

select dname from dept where deptno In (select deptno from emp where job = 'SALESMAN' group by deptno having count(*) >= 3);

realme Shot by ❤SANDIP❤

(9) display the dname of an employees who has no reporting manager.

Select dname from dept where deptno In (select deptno from emp where mgr is null);

(10) List all the employees who are reporting to Jones manager.

Select ename from emp where mgr In (select mgr from emp where ename = 'JONES');

(11) List employees from research & accounting having atleast 2 reporting.

Select ename from emp where deptno In (select deptno from dept where dname In ('RESEARCH', 'ACCOUNTING') group by deptno having count(*) >= 2);

(12) Display the departname of employee whose name does not starts with 'S' and salary between 1500 to 3000.

Select dname from dept where deptno In (select deptno from emp where ename Like '%S%' and (sal between 1501 and 2999));

(13) Display the location of an employee in ~~accounting~~ whose sal is minimum salary but sal is greater than 2000.

Select loc from dept where deptno In (select deptno from emp where sal In (select Min(sal) from emp where sal > 2000));

(14) Display the location of an employee in Accounting department.

Select loc from dept where dname = 'ACCOUNTING';

realme Shot by SANDIP 2024/02/11 12:45

(15) Display the department's location that is having greater than four employees in it.

```
select loc from dept
where deptno In (select deptno from
emp group by deptno having
count(*)>=4);
```

(16) write a query to display all the employee whose job not same as 'allen' and salary is greater than martin.

```
select ename from emp where job <> (select
job from emp where ename = 'ALLEN')
and sal> (select sal from emp where
ename = 'MARTIN');
```

(17) Display all the emps who is having location is same as Adam's manager.

```
select ename from emp where deptno
In (select deptno from emp
where empno In (select mgr from emp
where ename = 'ADAMS'));
```

(18) Display the job, manager number of employees who is working for 'Jones'.

```
select job, mgr from emp
where mgr In (select empno from
emp where ename = 'JONES');
```

(19) Display the employee names, higher date, commission of ford's manager.

```
select ename, hiredate, comm
from emp
where empno In (select mgr from emp
where ename = 'FORD');
```

(20) Display the number of emps who are getting salary less than the [illegible] manager.

realme
Shot by SAMDIP
realme 8i 2024 02 11 12:45

Select count(*) from emp
where sal < (select sal from emp
where empno In (select mgr from emp where
ename = 'BLAKE'));

(21) List employees who location in chicago and their commission is zero.

Select ename from emp
where deptno In (select deptno from dept
where loc = 'CHICAGO') and COMM=0;

(22) List emps who work for sales department and their salary greater than avg sal of their department.

Select ename from emp
where deptno In (select deptno from dept
where dname = 'SALES'
and sal > (select avg(sal) from emp));

(23) List employees who are working in research department and they are manager.

Select ename from emp
where deptno In (select deptno from dept
where dname = 'RESEARCH')
and deptno In (select deptno from dept where
job = 'MANAGER');

(24) Display department name of the employees who earn commission.

Select dname from dept
where deptno In (select deptno from emp
where comm is not null);

(25) Display department name of the employee who earn maximum salary and have no reporting manager.

realme Shot by ❤SANDIP❤
realme 8i 2024 02 11 12:45

Select dname from dept where deptno In
(select deptno from emp where sal
In (select max(sal) from emp
where mgr is null));

WAQTD the employees name, and total salary of each employees in each department, provided that each department has at least four employees.

```
select ename, deptno, sum (sal+nvl (comm,0))
as "total sal" from emp
group by deptno, ename
having count(*) >= 4;
```

realme Shot by ❤SANDIP❤
realme 8i 2024 02 11 12:45

① What is SQL Sandbox in SQL Server?

Solution: SQL Sandbox is a safe place in the SQL Server environment where untrusted scripts are executed. There are 3 types of SQL Sandbox:

① Safe Access Sandbox: Here a user can perform SQL operations such as creating stored procedures, triggers, etc. but cannot have access to the memory as well as cannot create files.

② External Access Sandbox: Users can access files without having the right to manipulate the memory allocation.

③ Unsafe Access Sandbox: This contains untrusted codes where a user can have access to memory.

② What is the Cartesian Product of the table?

Solution: The output of cross join is called a Cartesian Product. It returns rows combining each row from the first table with each row of the second table. For example if we join two tables having 15 and 20 columns the cartesian product of two tables will be 15X20 = 300 rows.

③ What is a Relationship? How many types of Relationships are there?

Solution: The relationship can be defined as the connection between more than one table in the database.

There are 4 types of relationships:

- One-to-One Relationship.
- Many-to-One Relationship.
- Many-to-Many Relationship.
- One-to-Many-Relationship.

④ What do you mean by Stored Procedures? How do we use it?

Solution: A Stored Procedures is a collection of SQL statements that can be used as a functions to access the database. We can create these stored procedures earlier

realme Shot by ❤SANDIP❤

before using it and can execute them wherever required by applying some conditional logic to them. Stored procedures are also used to reduce network traffic and improve performance.

> Syntax:

```
CREATE Procedure Procedure_Name
(
// parameters
)
As
BEGIN
SQL Statements in Stored procedures
                to update/retrieve records
END
```

⑤ What is Nested Triggers?

Solution: Triggers may implement data modification logic by using INSERT, UPDATE, and DELETE statements. These triggers that contain data modification logic and find other triggers for data modification are called Nested Triggers.

⑥ What is a Cursor?

Solution: A cursor is a database object which is used to manipulate data in a row-to-row manner.

The cursor follows steps given below:

- Declare Cursor
- Open Cursor
- Retrieve row from the cursor
- Process the row
- Close cursor
- Deallocate cursor

⑦ What are transactions and their controls?

Solution: A transactions can be defined as the sequence task that is performed on database in a logical manner to gain certain results. operations like creating, updating and deleting records performed in the database come from transactions.

In simple words, we can say that a transactio realme s [illegible] queries executed on database records.

realme 8i 2024 02 11 12:45

There are 4 transactions controls such as

- COMMIT: It is used to save all changes made through the transaction.
- ROLLBACK: It is used to roll back the transaction. All changes made by the transaction are reverted back and the database remains as before.
- SETTRANSACTION: set the name of the transaction.
- SAVEPOINT: It is used to set the point where the transaction is to be rolled back.

(8) What is Scalar function in SQL?

Solution Scalar functions are used to return a single value based on the input values.

(9) How many types of Privileges are available in SQL.

Solution There are two types of privileges used in SQL, such as

- System Privilege: System privilege deals with the object of a particular type and provides users the right to perform one or more actions on it. These actions include performing administrative tasks, ALTER ANY INDEX, ALTER ANY CACHE GROUP creates /ALTER / DELETE TABLE, CREATE /ALTER/DELETE VIEW etc.
- Object privilege: This allows us to perform actions on an object or object of another users(s) viz. table, view, indexes, etc. Some of the object privileges are EXECUTE, INSERT, UPDATE, DELETE, SELECT, FLUSH, LOAD, INDEX, REFERENCES, etc.

(10) What is collation?

Solution Collation is a set of rules that check how the data is sorted by comparing it. Such as character data is sorted using the correct character sequence along with case sensitivity, type, and accent.

realme Shot by ❤️SANDIP❤️
realme 8i 2024 02 11 12:45

(11) What do we need to check in Database Testing?

Solution In Database testing, the following thing is required to be tested:

1. Database connectivity.
2. Constraint check.
3. Required application field and its size
4. Data Retrieval and processing with DML operations.
5. Stored Procedures.
6. Functional flow.

(12) What is Database White Box Testing?

Solution Database white Box testing involves:

1. Database consistency and ACID Properties.
2. Database triggers and logical views.
3. Decision Coverage, Condition Coverage and Statement Coverage.
4. Database Tables, Data Model, and Database Schema.
5. Referential integrity rules.

(13) What is Database Black Box Testing?

Solution Database Black Box testing involves:

1. Data Mapping
2. Data Stored and retrieved
3. Use of Black Box testing techniques such as Equivalence Partitioning and Boundary Value Analysis (BVA)

(14) What are Indexes in SQL?

Solution The index can be defined as the way to retrieve data more quickly. We can define indexes using CREATE Statements.

Syntax:

```
CREATE INDEX index_name
ON table_name(Column_name)
```

realme Shot by ❤ SANDIP ❤
realme 8i 2024 02 11 12:46

Further, we can also create a unique Index using the following syntax:

CREATE UNIQUE INDEX index-name
ON table-name(column-name)

(15) What is identity in SQL?

Solution An identity column where SQL automatically generates numeric values. We can define a start and increment value of the identity column.

(16) Which TCP/IP port does SQL Server run?

Solution By default SQL Server runs on port 1433.

(17) What do you mean by query optimization?

Solution Query Optimization is a process in which a database system compares different query strategies and selects the query with the least cost.

(18) What is Referential Integrity?

Solution Set of Rules that restrict the values of one or more columns of the tables based on the values of the primary key or unique key of the referenced table.

(19) Define a temp table?

Solution A temp table is a temporary storage structure to store the data temporarily.

(20) What is Schema?

Solution A schema is a collection of database objects of a User.

(21) Does View Contain Data?

Solution No, Views are virtual structures.

(22) Can a View be based on another View?

Solution Yes, A view is based on another View.

(23) What is CTE?

Solution A CTE or common table expression is an expression [illegible] a temporary result set which is defined in a SQL [illegible]

realme Shot by SAMEER
realme 8i 2024 02 11 12:46

(24) What is a clause?

Solution A condition imposed on a SQL query to filter the data to obtain the desired result. Some examples are WHERE, LIMIT, HAVING, LIKE, AND, OR, ORDER BY etc.

(25) What is the diffn between local and global variables?

Solution Local variables can be accessed only inside the function in which they are declared. Instead, global variables, being declared outside any function, are stored in fixed memory structures and can be used throughout the entire program.

(26) Can we create a view based on another view?

Solution Yes, This is also known as nested views. However, we should avoid nesting multiple views since the code becomes difficult to read and debug.

(27) Can we still use a view if the original table is deleted?

Solution No. Any views based on that table will become invalid after deleting the base table. If we try to use such a view anyway, we'll receive an error message.

(28) What types of SQL relationships do you know?

Solution
- One-to-One - each record in one table corresponds to only one record in another table.
- One-to-many - each record in table corresponds to several records in another table.
- Many-to-many - each record in both tables corresponds to several records in another table.

(29) What is field?

In DBMS, a database field can be defined as - a single piece of information from a record.

realme Shot by SANDIP
realme 8i 2024 02 11 12:46

(30) What is a Default Constraint?

Solution: The default constraint is used to fill a column with default and fixed values. The value will be added to all new records when no other value is provided.

(31) What is the On Delete cascade Constraint?

Solution: An 'On Delete cascade' constraint is used in MySQL to delete the rows from the child table automatically when the rows from the parent table are deleted.

(32) What is Denormalization?

Solution: Denormalization is a database optimization technique in which we add redundant data to one or more tables. This can help us avoid costly joins in a relational database.

Note that, denormalization doesnot mean not doing normalization. It is an optimization technique that is applied after normalization.

In a traditional normalized database, we store data in separate logical tables, and attempt to minimize redundant data. We may strive to have only one copy of each piece of data in the database.

(33) What is Cursor?

Solution: The cursor is a Temporary Memory or Temporary Work Station. It is Allocated by Database Server at the Time of performing DML operations on the Table by the User. Cursors are used to store Database Tables.

(34) What is Trigger?

Solution: A DB trigger is a code or programs that automatically execute with response to some event on a table or view in a database. Mainly, trigger helps to maintain the integrity of the database.

Example: when a new Student is added to the Student database, new records should be created in the related tables Exam, Score, and Attendance tables.

realme Shot by realme 8i 2024 02 11 12:46

(35) What is Auto Increment?

Solution Auto Increment Keyword allows the user to create a unique number to be generated when a new record is inserted into the table. Auto Increment Keyword can be used in Oracle and IDENTITY Keyword can be used in SQL SERVER.

Mostly this Keyword can be used whenever PRIMARY KEY is used.

(36) What is collation?

Solution Collation is defined as set of rules that determine how character data can be sorted and compared. This can be used to compare A and, other language characters and also depends on the width of the characters.

(37) What are all different types of collation sensitivity

Solution Following are different types of collation Sensitivity:

- Case sensitivity - A and a and B and b
- Accent Sensitivity.
- Kana Sensitivity - Japanse Kana characters
- Width Sensitivity - Single byte character and double byte character.

(38) What is query?

Solution A DB query is a code written in order to get the information back from the database. Query can be designed in such a way that it matches with our expectation of the result set. Simply a question to the Database.

(39) Advantages and disadvantages of Stored Procedure?

Solution Stored Procedure can be used as a modular Programming means create once, store and call for several times whenever required. The supports faster execution instead of executing multiple queries. This reduces network traffic and provides better security to the data.

Disadvantage is it can be executed only in the Database and utilises more memory in the database server.

realme 8i 2024 02 11 12:46

(40) What is online Transaction Processing (OLTP)?

Solution: Online Transaction Processing (OLTP) manages transaction based applications which can be used for data entry, data retrieval and data processing. OLTP makes data management simple and efficient. Unlike OLAP systems goal of OLTP systems is serving real-time transactions.

Example: Bank Transactions on a daily basis.

(41) What is recursive stored procedure?

Solution: A stored procedure which calls by itself until it reaches some boundary condition. This recursive function or procedure helps programmers to use the same set of code any number of times.

(42) What are SQL comments?

Solution: Comments are explanations or anotations in SQL queries that are readable by programmers. It's used to make SQL statements easier to understand for humans. During the parsing of SQL code, it will be ignored. Comments can be written on a single line or across several lines.

* Single Line comments: It starts with two consecutive hyphens (--).
* Multi-line Comments: It starts with /* and ends with */.

(43) What is a data Warehouse?

Solution: A data Warehouse is a large store of accumulated data, from a wide range of sources, within an organization. The data helps drive business decisions.

(44) Mention different types of replication in SQL Server?

Solution: In SQL server, three different types of replications are available:

- Snapshot replication
- Transactional replication
- Merge replication

realme Shot by SANDIP
realme 2024.02.11 12:46

(45) Which command is used to find out the SQL Server Version?

Solution: The following Command is used to identify the version of SQL Server:

Select SERVERPROPERTY('productVersion')

(46) Explain the difference between spreadsheets and database?

① Spreadsheet: A file that exists of cells in rows and columns and can help arrange, calculate and sort data is known as Spreadsheet. It can have a numeric value, text, formulas and functions. It features columns and rows to keep inserted information legible and simple to understand. It is an electronic graph sheet.

> Example: Microsoft Excel, Lotus 1-2-3.

② Database: It is an organized collection of data arranged for ease and speed of search and retrival. It contains multiple tables. A database engine can sort change or serve the information on the database. Basically, it is a set of information which is held in a computer.

> Example: Banks use databases to keep track of customer accounts, balances and deposits.

Spreadsheet	Database
① It is accessed directly by the user.	① It is accessed by a user or by an application to enter or modify the data.
② It can be classified as numeric relationships.	② It can be classified as classification of data.
③ It is easy to learn for the user.	③ It is somewhat difficult than spreadsheet to learn.

realme Shot by SANDIP
realme 8i 2024.02.11 12:46

④ It is an interactive computer application for organization, analysis and storage of data in tabular form.	④ It is an organized collection of data generally stored and accessed electronically from a computer system.
⑤ It stores less data.	⑤ It stores more data.
⑥ It is used for accounting tasks.	⑥ It is used in large enterprises to store the data.
⑦ It have more formatting features than database.	⑦ It automatically update forms, reports and queries when data is updated.

(47) What is data integrity?

Solution: Data Integrity in SQL refers to the presence of true and correct data in a database. It ensures that the data is accurate reliable and consistent throughout its life cycle, including data collection, storage, changes, migrations, restorations etc.

In SQL there are 4 types of data integrity:

① Domain integrity: The authenticity of inputs for a particular column is called domain integrity in SQL.

② Entity integrity: Entity integrity requires that every row in the table should have distinct records.

③ Referential integrity: A foreign key table in a database is a second table that may refer to a database's primary key.

④ User-defined integrity: User-defined integrity is a set of rules that the user can define to ensure data integrity.

realme Shot by ❤SANDIP❤

Constraints can be used to maintain data integrity in a SQL database. The constraints describe the rules that must be followed while doing action such as updating, deleting and inserting data to maintain data integrity.

realme Shot by ❤️SANDIP❤️
realme 81 2024 02 11 12:47

Difference betn WHERE Clause and HAVING Clause:

WHERE	HAVING
① Where clause can be used with or without using GROUP BY Clause.	① Whereas HAVING clause can only be used with GROUP BY clause.
② It is used in row operations.	② It is used in column operations
③ It cannot use aggregate functions (sum, avg, count etc). the only exception is when it is a part of the HAVING clause.	③ It can use aggregate functions.
④ It can be used with all DML Commands (INSERT, UPDATE, DELETE).	④ It can only be used while querying using a SELECT statement.
⑤ It can perform only a single row function.	⑤ It can perform multiple row functions.

Difference betn HAVING clause and GROUPBY clause

Having clause	GroupBy clause
① It is used for applying some extra condition to the query.	① The groupby clause is used to group the data according to particular column or row.
② Having cannot be used without groupby clause, in aggregate function, in that case it behaves like where clause.	② groupby can be used without having clause with the select statement.

HAVING clause	GROUPBY clause
③ The having clause can contain aggregate functions.	③ It cannot contain aggregate functions.
④ It restrict the query output by using some conditions.	④ It groups the output on basis of some rows or columns.

Difference between GROUPBY clause and ORDER BY clause

GROUP BY clause	ORDER BY clause
① It is used to group the data on the basis of particular column.	① It's used to display the information either in ascending order or in descending order.
② It is mandatory to use aggregate functions in the GROUPBY.	② It's not mandatory to use aggregate functions in the ORDER BY.
③ It is always used before the ORDER BY clause in SELECT statement.	③ It is always used ~~for~~ after the GROUP BY clause in the SELECT statement
④ It controls the presentation of rows.	④ It controls the presentations of columns.

Difference between subquery and co-related subquery.

Difference between BLOB and CLOB

BLOB	CLOB
① The full form of blob is Binary Large Object.	① The full form of clob is Character Large Object.
② This used to store large binary data.	② This is used to store large textual data.
③ This stores values in the form of binary streams.	③ This stores values in the form of character streams.
④ using this you can store files like text files, PDF documents, word documents.	④ using this you can stores files like videos, images, gifs, and audio files.

Difference between Nested subquery and correlated subquery

Nested subquery	correleted subquery
① In Nested query, a query is written inside another query and the result of the inner query is used in the execution of the outer query.	① In correleted query, a query is nested inside another query and and an inner query uses values from the outer query.
② Bottom-up approach i.e. inner query runs first, and only once. The outer query is executed with result from the inner query.	② Top to Down Approach i.e. outer query executes first and for every outer query row Inner query is executed.
③ Inner query execution is not dependent on outer query.	③ Inner query is dependent on the outer query.

Nested Subquery	Correlated subquery
④ Performs better than Correlated query but is slower than Join operation.	④ Performs slower than both Nested query and join operation as for every outer query inner query is executed.

Difference between Single Row Function and Multi-Row Function

Single Row Function	Multi Row Function
① Single Row Func operate on a single row and return one result per row.	① Multi Row Func operate on a group of rows and return one result for the entire group.
② Single Row Func are used to manipulate data within a row.	② While Multi Row Func are used to perform calculations on a set of rows.
③ Single Row Func are used in the SELECT clause of a query.	③ Multi Row Func are used in the SELECT clause along with the GROUP BY clause.
④ Single Row Func can be nested within other single row or multi Row functions.	④ Multi Row Functions can not be nested within other multi row functions.
⑤ Single Row Func can accept and return various data types including numbers, characters and dates.	⑤ Multi Row Functions typically only accept and return numeric data types.

Single Row Functions	Multi Row Functions
⑥ Single Row Functions can handle null values in various ways depending on the functions.	⑥ While Multi Row Functions typically ignore null values when performing calculations.
⑦ Examples of Single functions include UPPER, LOWER, TRIM, SUBSTR and ROUND.	⑦ Examples of Multi Row Functions include SUM, AVG, MIN, MAX and COUNT.

Difference between RowId and RowNum

RowId	RowNum
① Physical address of the rows.	① Rownum is the sequential number, allocated to each returned row during query execution.
② Rowid is permanant	② Rownum is temporary.
③ Rowid is 16-bit hexadecimal	③ Rownum is numeric.
④ Rowid gives address of rows or records.	④ Rownum gives count of records.
⑤ Rowid is automatically generated unique-id of a row and it is generated at the time of insertion of row.	⑤ Rownum is an dynamic value automatically retrieve along with select statement output.
⑥ Rowid is the fastest means of accessing data.	⑥ It represents the sequential order in which oracle has retrieved the row.
⑦ They are permanent unique identifiers for the any row in a table.	⑦ Rownum is temporary and changes based on the specific SQL statement.

Diffⁿ betⁿ clustered Index and Non clustered Index

Clustered Index	Non clustered Index
① The clustered index is faster.	① The Non-clustered Index is slower.
② The clustered index requires less memory for operation.	② The non-clustered index requires more memory for operations.
③ In a clustered Index, the index is the main data.	③ In the Non-clustered index, the index is a copy of data.
④ A table can have only one clustered index.	④ A table can have multiple non-clustered indexes.
⑤ In the clustered index, the clustered key defines the order of data within the table.	⑤ In the Non-clustered index, the index key defines the order of data within the index.
⑥ A clustered index is a type of index in which table records are physically reordered to match the index.	⑥ A Non-clustered index is a special type of index in which the logical order of index does not match the physical stored order of the rows on the disk.

Diffⁿ betⁿ SQL and PL/SQL :

SQL	PL/SQL
① It is a database Structured Query Language.	① It is a database programming language using SQL.
② Variables are not available in SQL.	② Variables, constraints and data types features are available in PL/SQL.
③ No supported control structures like for loop, if and other.	③ Control structures are available like for loop, while loop, if and other.

SQL	PL/SQL
④ It is a Data oriented language.	④ It is an application oriented language
⑤ SQL is a declarative language.	⑤ PL/SQL is a procedural language.
⑥ SQL can be embedded in PL/SQL.	⑥ PL/SQL can't be embedded in SQL.
⑦ It directly interacts with the database.	⑦ It doesn't interact directly with the database server.
⑧ SQL does not provide error and exception handling	⑧ PL/SQL provides error and exception handling.
⑨ It is used to write queries using DDL and DML statements.	⑨ The code blocks, functions, procedures triggers and packages can be written using PL/SQL.
⑩ You can fetch, alter, add, delete or, manipulate data in a database using SQL.	⑩ You can use PL/SQL to develop applications that show information from SQL in a logical manner.

Diff^n betn SQL, MySQL and SQL server

SQL	MySQL	SQL server
① SQL or Structured Query Language is useful for managing our relational database. It is used to query and oporate the database.	① MySQL is the popular database Management System used for managing the relational database. It is a fast, scalable, and easy-to-use database.	① SQL server is an RDBMS database System mainly developed for the windows System to store retrieve, and access data requested by the developer.

SQL	MySQL	SQL Server
② SQL was developed by IBM Corporation.	② MySQL was developed by Oracle Corporation.	② SQL server was developed by Microsoft company.
③ SQL is a query language for managing database.	③ MySQL is database software that uses SQL language to conduct with the database.	③ SQL server is also a software that uses SQL language to conduct with the database.
④ SQL has no variables.	④ MySQL can use variables constraints and data types.	④ SQL server can use variables constraints and data types.
⑤ SQL is a programming language, so that it does not get any updates. It's commands are always fixed and remain the same.	⑤ MySQL is software, so it gets frequent updation.	⑤ SQL server is also software, so, it gets frequent updation.