

RDBMS Concepts

DB- stands for database

Database

- Storage of data, piece of useful information
- Storage of data in the form of tables.

* Data is a useful Info.

- A data is regarded as processed info on which some decision is possible.
- Data is used for making Some business Analysis.

* Information - is unprocessed data, it may or may not be useful.

DATABASE

- The place of Storage of the data in the form of tables.
- DB is also known as "collection of tables"
- is a Systematic organization of an useful information (data) in the form of tables.

* A db is simply stated as collection of one or more tables

↓ ↓ ↓ columns

X	Y	Z
-	-	-
-	-	-

cells

Table: is a collection of rows & columns
rows & A table is also called as entity or a relationship

* Other way to store data is through Files

Column: Also Known as field or Attribute

- * It represents the property of a table.
- * In Oracle a table can have upto 1000 columns (Minimum of 1) and 'n' no of tables
- * Columns can not have same Name in a column.

Row: Also known as record or tuples

- * It represents the data
- * we can have unlimited no of records depending on db size (hard disk), upto the size of the harddisk

CELL: is the intersection of rows & columns

- * A column @ a row is always a collection of cells.

→ To read or write the data in the table, in the form cells.

→ In RDBMS we always store/retrieve the data in the form cells.

DB → Like client / Server architecture

Advantage: Security

Better performance

Time to retrieve data

Types of DB

* DBMS - Database Management System

* RDBMS - Relational db mgmt System [Oracle, SQL, MySQL, DB2]

* ORDBMS - Object-Relational database management System

DBMS! ^{helps to} Store the data in the form of tables.



DB SW! It allows us to create the db which can store the data in the form of tables.

* When a db sw is installed a portion of hard disk is reserved to store the data in the form of tables which is called as db. (Reside in the hard disk)

* The size of the db is directly proportional to the size of the hard disk.
* The DB size depends on the hard disk size, we can extend the db size till the size of the hard disk.

DBMS

1. has less features

2. will not support Codd-Rules

3. Lesser performance

4. Less Storage Capacity

5. Not So Stable

6. Not Scalable

7. Less Security

* Any DBMS to qualify as RDBMS → It should satisfy "CODD Rules".

RDBMS

1. has more features

2. Supports minimum 5-6 Codd(Rules)

3. Good performance

4. More Storage Capacity

5. Stable

6. Scalable

7. More Security

Sq1 DBMS ^{CODD} Rules RDBMS

Fox base

Oracle, DB2, Informix, projects

Fox pro

SQL server, Teradata, SAP hana

DBase

Sybase

Lotus

My SQL

In ORDBMS we can perform object oriented design & programming

* In DBMS when we create a table a separate file will be created for each table

* In RDBMS - One file contains multiple tables.

* RDBMS has better features over DBMS in terms of better storage, better security

Supports all RDBMS features & also some object oriented features.

Sq1. Oracle 11g.

- Find out 5 different DBMS & RDBMS
- Mention few Codd Rules

ASSIGNMENT

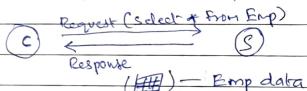
File Server v/s Client Server.

File Server: In file server architecture the entire data is pumped across the network even though we requested only part of it. This results in performance issues.

Ex: Note pad, Excel Sheet, Word pad

Client Server: In client server architecture we will get only data (response) for which we requested for.

Ex: Database Server, RDBMS, internet & intranet applications.



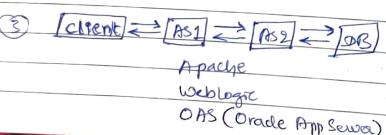
Types of Client Server Architecture

- ① 2-tier
- ② 3-tier
- ③ n-tier

① 2-tier



② 3-tier

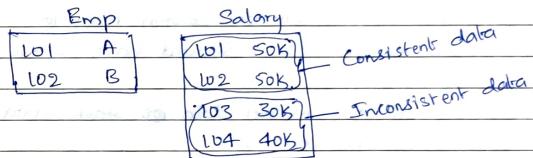


Relationships:

is the association b/w any two entities to maintain the data consistency (Data Integrity)

Independent tables — Master tables

Dependent tables — Child tables



* By the relationship we can avoid inconsistent data.

Assignment

- ① Give an example for relationship is mandatory
- ② Find out the different types of Relationships.

Institute		Admission detail	
Course	Fees	Student Name	Course
PLSQL	6000	A	Java
JAVA	8000	B	.Net
.Net	6000	C	PLSQL

Relationship		Emp	
Deptno	Dept Name	EmpNo	EmpName
10	Account	101	John
20	Sales	102	Sri
30	HR	103	Dix
	!	104	Rai

Deptno

Dept Name

EmpNo

EmpName

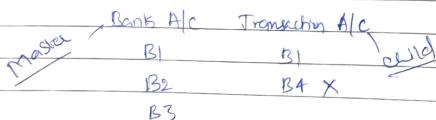
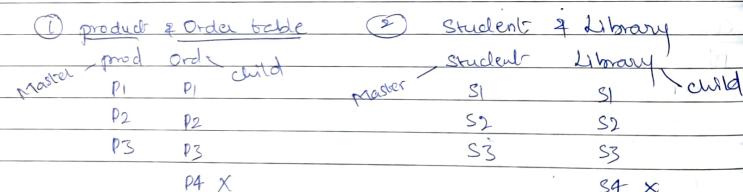
Rejected

child/parent
dependent table

Master/parent
Independent tables

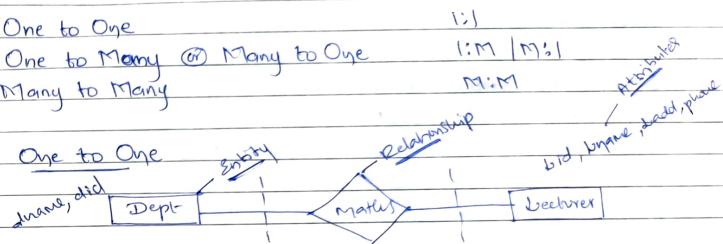
- * When we create a relationship, one table becomes Master table / parent table and other becomes child / detail table.
- * Any data present in the child table should be present in the Master table, otherwise the data in the child table is rejected.
- * To Create a relationship the 2 tables should have common columns between them. The common column may or may not have same name but it should have same data type.
- * The master table contains the master data (static data) which may not change frequently.
- * The child table contains the transactional data (more volatile data than the Master)
- * The child table generally have foreign key

Note: The relationship can be created by creating the foreign key.

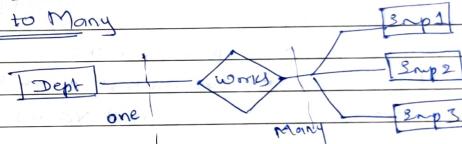


Types of relationships

- ① One to One
- ② One to Many
- ③ Many to One
- ④ Many to Many

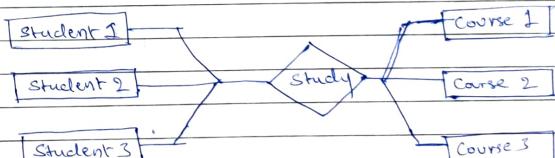


- ② One to Many



Ex: College - Students, Organisation - Employees

- ③ Many to Many



One student can study many courses / One course can be studied by many students.

Ex: Student - Subjects, bank - Customers

ASSIGNMENT

- ① Write atleast 5 examples for each type of Relationship.

NULL Values :-

Eid	Ename	Deptno	Comm
101	A	10	222*
102	B	20	0
103	C	0	5555
104	D	0	0

- * A null is empty value @ no values. @ undefined values
- * A null is neither zero nor blank space
- * It represents unknown @ uncertain value @ missing value @ indefinite values
- * A null is a '0' (zero) byte character.
- * A null will not occupy any space in memory. (uniquely)
- * In oracle db, two null values are always treated differently
- * Any comparisons done against the null values should always result in false
- We can't do compare null values during uncertainty
- * Two nulls are not same
- * Any arithmetic operation performed in null will result in null itself. → $comm + 200$

→ Ex!:

	Comm > 2000	Comm < 2000	Comm ≠ 2000
A	T	F	422
B	F	P	0
C	T	F	785
D	F	P	0

→ In Sql Server db, two nulls are always treated as same.

- * If we want to avoid null values in a field, we should give not null constraints.

CONSTRAINTS

A constraint is a condition or a limitation which is enforced on the column of the table to avoid the incorrect data.

- * We can have more than one constraint on the column.
- * Constraints are given on column of a table.
- * A constraint enabled @ disabled (status)
- * Enabled constraint checks for the conditions (attribute), disabled constraint should be present but not activated (They will not check).

Empno	Ename	Sal	Deptno
101	A	*	10
101	B	+	20
102	C	-6000	30
105		00*	10
108	D	*+	40

Types of Constraints

1. Not null
2. Unique
3. Primary Key
4. Foreign Key
5. Check
6. default

1. NOT NULL

- A column should have some values
- It will not allow any null values.
- It marks a column as mandatory.
- We can have 'n' no of NOT NULL's in a table

2. UNIQUE

- It will not allow duplicate values
- We can have 'n' no of unique columns
- It will allow multiple null values in unique column (Oracle)
(One null is != another null)
- Sgl db - all null are same, hence we can have only one null in Unique column.

3. Primary Key

- Uniquely identifies a row in a table.
- PK is used to identify the unique record in the given table.
- Only one primary key for One table
- PK = Unique + Not Null

PK →	EmpNo	Enname	Sal	Dno	Phone	Email	Unique + NN
	101	Smitha	4000	10	111	abc@gmail	
	102	Nicety	6000	20	222	xyz@gmail	

- * It is not mandatory to have a primary key, but it is highly recommended to have it.
- * If we want to create multiple PK, One column made as PK and ^{PK} others should be given as Unique + Not Null

ASSIGNMENT

- ① Write atleast 2 differences b/w PK & Unique Key
- ② What is the difference b/w PK & Unique + NN Combination.

PK

- ① Only one per table
2. No null values
3. Never allows the duplicates

UNIQUE

- multiple unique keys
- multiple nulls
- Not allow duplicated (Except nulls)

PK

- One in a table

Unique + Not Null

- 1 or More in a table

Types of primary keys.

1. Simple primary key
2. Composite primary key
3. Surrogate primary key.

1. Simple Primary Key

If a primary key is made of only one column is called as Simple PK.

2. Composite primary key.

If a primary key is made of more than one column then it is called as Composite PK.

* We can opt to create Composite PK when we cannot create Simple PK.

We can have upto 16 columns for a Composite PK making

PK Tempno	Ename	Job	PK		
			Eno	Mon	Year
101	A	Mgr	101	Jan	2012
102	B	SM	101	Feb	2012
103	C	Mgr	102	Jan	2012
104	D	SM	102	Feb	2012
105	E	Mgr	101	Jan	2013

$$\text{here } \text{PK} = \{\text{Eno} + \text{Mon} + \text{Year}\}$$

So it is known as Composite PK

ASSIGNMENT

* Generally we create numerical fields for the PK, for accessing number is fast

- (1) Write 2 scenarios for Composite primary key
- (2) Identify 2 tables in a healthcare domain & put appropriate constraint for them.

3. Surrogate primary key.

Surrogate PK Transaction table

TransId	Acct no	Amnt	Date	Type
1	101	2000	1-nov-13	D
2	101	3000	1-nov-13	D
3	102	2000	1-dec-13	D

* Surrogate Key is like a artificial column created just for the purpose of Primary Key

* Surrogate Key can be generated using sequences in Oracle

* In SQL Server we use 'identity' property for the same.

* It is useful when we are not able to create either simple / composite key

→ IX! Transaction ID in the banking application

→ EX! Reference no, Acknowledgement #, PNR No, tracking no's

→ It contains generally running numbers or serial numbers [1, 2, 3, 4, 5, ...]

→ May or may not have business significance.

CANDIDATE & ALTERNATE KEYS



$$CK = AK + PK$$

$$PK = Eno$$

$$AK = CK - PK$$

$$AK = phone + Mail$$

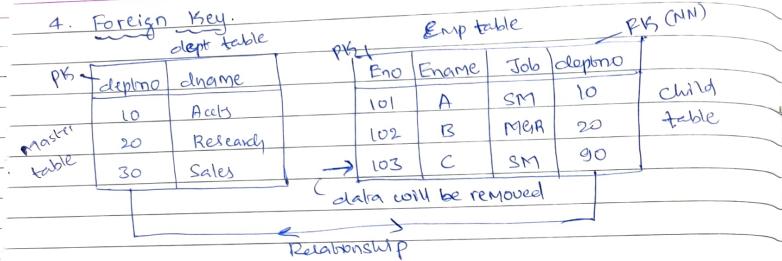
$$PK = CK - AK$$

$$CK = Eno + phone + Mail$$

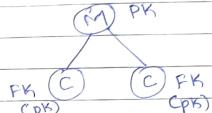
* The columns that are eligible to become PK are called as Candidate Keys

* The columns that are eligible to become PK but not made as PKs are called as Alternate Keys.

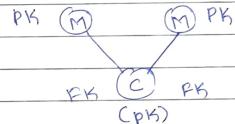
4. Foreign Key.



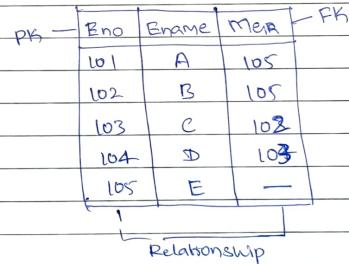
(b) A master can have multiple child tables.



(c) A child can be made up of multiple Master tables.



(d) A table can be self referenced to itself (PK ≠ FK within the same table but on different columns.)



Scenarios of FK

(a)



A master can have a child, which intern can have another child



(Pis) Not Compulsory

ASSIGNMENT

- ① Write atleast 2 examples for all the Scenarios
- ② Write 5 differences b/w PK & FK
- ③ Difference b/w Master & child table
- ④ Difference b/w Simple Key & Composite Key
- ⑤ What is Entity Relationship Diagram

* If we create a foreign key from multiple columns it is called as composite foreign key, It should be referenced by composite PK.

⑤ Check Constraint

Check is used to provide some business validations as per customer requirements

Ex: ① Sal > 0

② Age b/w 25 to 35 years

③ Gender M or F

④ Status 'A' or 'I'

⑤ Empid Should Start with 'EID'

⑥ Bonus <= 5000

* Check constraints can be used to implement only the simple rules OR conditions, Any complex rules that cannot be implemented achieved using check constraints can be implemented using triggers.

Limitations

→ In check constraints, we cannot use functions in the predefined OR user defined

→ we cannot use multiple columns in a single check constraints

⑥ Default.

* It provides default values to a column

* It will be taken only if the value is not entered, if we enter a value explicitly then that value is taken instead of default.

* In default we can use functions.

* If we provide a value during the insertion then default is not used.

Examples:

a. Commission default 0 , Tax default 0

b. Create date default Sysdate

c. DOJ default Sysdate

d. Country ID default IND , Status default 'I'

ASSIGNMENT ANSWERS

→ Simple Key - Made of Single Column

Composite Key - Multiple Columns

PK, FK, UK only composite where as Not null, Checks, default, PK, FK, UK can be simple.

→ Master table - PK , Unique Values, Not Null

child table - FIs & may or may not have PK, duplicate values, Null

Continent | country | state

University | College | Hospital | students

Doctor | College | Hospital

ASSIGNMENT

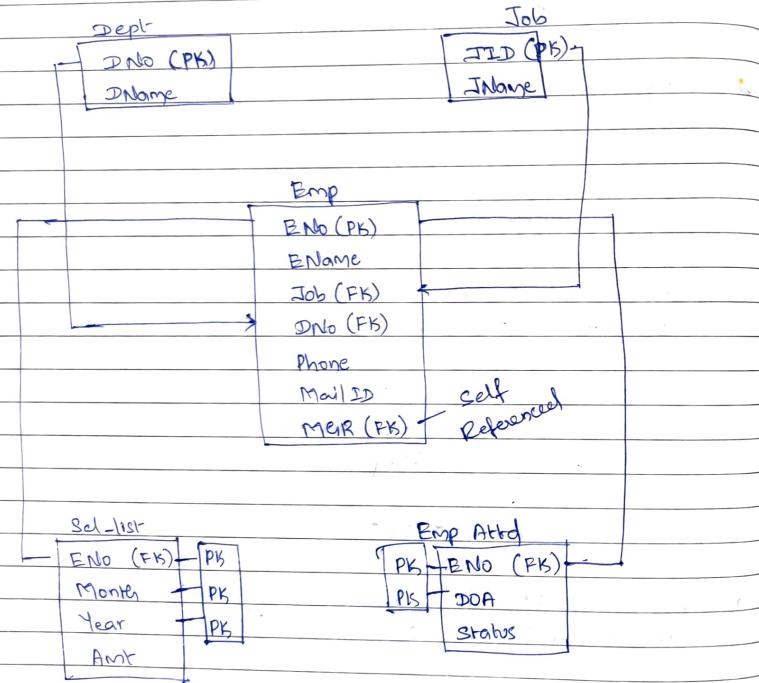
① Design an ERD for a healthcare application which covers all the FK constraints (6 tables)

② What is composite foreign key, with example (Created from composite primary key)

ERD - Entity Relationship Diagram

An ERD is the pictorial representation of the entities and their relationship.

It is the blue print / design document of the db.



One Child Multiple Master - Emp, dept, Job

One Master Multiple Child - Emp, Emp Attrd, Emp SelList

* A Single column can act as both PK & FK

ASSIGNMENT ANSWER

Composite foreign Key

Create table accounts (

acc-num Integer,

acc-type Integer,

acc-descr char(20),

Constraint PK acc Primary Key (acc-num, acc-type);

Create table Sub-accounts (

sub-acc Integer Primary Key,

(@) primary key (sub-acc, ref-num, ref-type)

ref-num Integer Not Null,

ref-type Integer Not Null,

Sub-descr Char(20),

Constraint FK_Subacc Foreign Key (ref-num, ref-type)

references accounts (acc-num, acc-type);

(@) Foreign Key (ref-num, ref-type) references accounts

A referential constraint must have a one-to-one relationship b/w referencing & referenced column.

If the primary key is a set of columns (a composite key) then the foreign key also must be a set of columns that corresponds to the composite key.

CPI Category (Master)

col1 to col3

CPI Name

lack (col)

NICEPI (child)

col1 to col7

CPI Name

TBC Code

Health Care Application - ERD

Patient

Pat-id (PK)

P-Name

Rx-id (FKS)

Gender

DOB

Addl

Prescriber

Pr-id (PK)

Pr-Name

Pat-id (FKS)

Rx history

Rx-id (PKS)

item-id (FKS)

item-name

Status

Date/time

pharm-id (PKS)

pre-id (FKS)

Pat-id (PKS)

Pharmacist

ID (PK, FKS)

Name

Addl

pharmacy Staff

id (PK)

Name

Designation

item

item-id (PKS)

item-name

QOH

QOH



other tables

Hospital
Doctor
patient & history
Drugs
Orders

Patient

Pat-id (PK)

P-Name

Gender

DOB

Addl

Rxid

Rx Suffix](FKS)

Prescriber

id (PK)

rname

Pat-id -(FKS)

Item

id (PKS)

Name

QOH

NDC

Rx history

Rxid](PKS)

Rx Suffix](FKS)

itemid (FKS)

Status

Date/time

pharm-id (FKS)

pre-id (FKS)

pat-id (PKS)

pharmacy Staff

id (PK)

Name

Role

Prescriber

id (PK)

Name

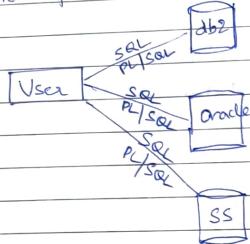
Pat-id (FKS)

Pharmacist
Name
Addl

SQL

SQL stands for Structured Query Language

It is a language to talk to the db, in order to store
retrieve from the db, we need to make use of the SQL



Any RDBMS Should
support SQL language
compulsorily

SQL is case insensitive language, is also called as non
procedural language, is also called as 4th Generation
language @ 4GL

The procedural extension of SQL is called as PL/SQL
wherein we can write procedural statements like programs
involving IF statements, Loops etc., along with SQL stats.

DB: (oracle, SQL Server, DB2, Sybase)

Oracle (oracle db)

12C - cloud

11g
log
grid

9i
8i
Intelligent

8

7

6

5

4

3

2

Default tables created while installing Oracle

Emp

dept

User

Connecting to the db : scott/tiger (SQL Plus)
Password is case-sensitive

SQL> show user;

SQL> select * from v\$version; (db Version)

SQL> select * from global_name; (db Name)

SQL> select * from tabs; (db tables)

SQL> desc dept; (table Structure)

SQL> select * from dept;

* Used to retrieve all records (* ALL columns)

SQL> select dname, loc from dept; (selected columns)

SQL> set linesize 120; (Set characters per line)

SQL> select * from emp;

SQL> set pagesize 120; (Set no of lines per page)

SQL> select * from emp;

SQL> clear;

Note : Single quotes for DATE & Char
Oracle doesn't sort the data

WHERE CLAUSE

It is used to filter the records being displayed

Select * from Emp where deptno=20;
—
— where Job = 'MANAGER'
—
— where Job = 'manager'

→ Commands are not case sensitive but data present in it
are case sensitive

→ strings will be provided within single quotes ''
and date will be provided within (' ') Single quotes
where hiredate = '01-dec-14'

SORTING - "ORDER BY" clause

→ Arranging data in ascending @ descending order
→ Sorting can be done by using "ORDER BY" clause
→ By default it arranges the data in the ascending order

Select * from Emp where deptno=20
order by Empno.

* We can also use column numbers instead of column name

Select * from Emp Order by ⑤ desc;

column 5 from Emp table will be considered

* Column name - need not be part of Selectr statement-
Column Number - Should be part of Selectr Statement-

Selectr Ename, Sal, Job, deptno from Emp
where deptno=20
Order by 2;
(2 = Sal)

Order by 5; — gives error
Order by Empno — works

* Sorting using multiple columns

Selectr * from Emp order by deptno, Job;

Select Statement Syntax

Select < col1, col2, ... coln > } Mandatory
 From < Table Name >
 Where < non-aggregate condition >
 Group by < col1, col2, ... coln >
 Having < aggregate condition >
 Order by < col1, col2, ... coln >

Selecting distinct values / Unique values

Select distinct Job from Emp;

Select Unique Job from Emp;

Select distinct deptno, Job from Emp;

Select deptno, distinct Job from Emp; → Results in Error

Distinct should be used always at the beginning of a column or all columns

ASSIGNMENT

1. List all the managers & sort the data according to their salaries

⇒ Select * from Emp

Where Job = 'MANAGER'

Order by Sal;

2. What is the department for dept no 20

⇒ Select dname from dept

Where deptno = 20;

3. List all the clerks and arrange the data according to hire date
4. Display the unique jobs from dept = 30
5. Sort the data in desc order
6. Display all the emp from dept 30 and sort the data in the desc of commission
7. List the emp who are earning more than 2K
8. List the emp except managers
9. List the no of emp in dept 10
10. display the department located in newyork

SQL OPERATORS

1. Relational Operators ($>$, $<$, \geq , \leq , $=$, \neq , $<>$, \wedge)
2. Arithmetic Operators ($+$, $-$, $*$, $/$)
3. Logical Operators (OR, AND, NOT)
4. Special Operators (IN, LIKE, BETWEEN, IS)
5. Concatenation Operator (||).
6. Set Operators (UNION, UNION ALL, Intersect, minus)

Relational Operators

Select * from Emp
where sal > 200;
Job != 'MANAGER';
Deptno = 20;

Special Operators

IN Operator: is used for comparing multiple values

Select * from Emp where deptno IN (20, 30);

Select * from Emp

where ename like 'A.%' - name starts with A

'%.A' - name ends with A

'%A.%' - name having letter 'A' anywhere

' - O.%' - name having 2nd character 'O'

Sal like '2.%'

hiredate like '%DEC-%'

Ename like 'man.%' '%man'

10g onwards } Google Regular Expression
REGEXP-like

BETWEEN Operators:

It reaches data based on range values

It returns the range of values

Select * from Emp

where Sal between 2000 and 3000;

where hiredate between '01-JAN-1980' AND '31-DEC-1980'

Inclusive

Exclusive

when we use characters, we are suppose to give one plus

if we want to display names from 'A' to 'F', then
we are suppose to give 'A' to 'E'

* To display Name from 'A' to 'Z'

Select ASCII('A') from dual;

$$\text{ASCII ('A')} = 65$$

Select CHAR(65) from dual;

$$\text{CHAR}(65) = A$$

$$\text{ASCII ('Z')} = 90$$

$$\text{CHAR}(90) = Z$$

Select * from Emp

where Ename Between 'A' and 'I';

where Ename Between 'I' and CHAR(ASCII('Z')+1)

IS operators! is used for comparing only NULL values
Empty values.

Select * from Emp

where COMM IS NULL;

O/P: Displays the values

Select * from Emp

where COMM = NULL;

O/P: Error because oracle does not differentiate 2 Nulls

B. Arithmetic Operators: [+,-,*,/]

Hierarchy: (), *, /, +, -

[Hierarchy can be changed using ()]

Select Ename, Sal as "currSal", Sal+2000 as NewSal,
Sal + (Sal*0.2) as NewSal2 from Emp;

If we add any column with null value results in null value

Select Ename, Sal, COMM, Sal + NVL(COMM, 0) as tsal
from Emp;

Dual table (dummy table)

Dual is a dummy table, which is used for performing non-database operations [To perform independent operations @ calculations they are not dependent on any existing tables]

It is a system table which has one row and one column.
This table is used for Syntax purpose

Select SYSDATE from dual;

Select 20+90 from dual; dual is Mandatory in Oracle but not in SQL Server

Select * from dual;

Select User from dual;

Select Syntax

Select --- from ---

~~Oracle~~ From --- ~~SQL Server~~ Select --- from ND(Mandatory)

→ desc dual;

Airthmetical Operator Precedence

()

*] same level

+] same level

ASSIGNMENT

1. List Employee name, salary , 20% more sal as his new salary for all the Employees from deptno=20 and Sort the data according to new salary.

Select Ename, Sal, Sal + (Sal * 0.2) as "New Sal"
 From Emp where deptno = 20
 Order by "New Sal"
 Order by 3
 Order by Sal + (Sal * 0.2)

Alias will be used only in Select Statement &
Order by Clause

2. List all the employees whose total Salary is more than 1500 (Total Sal = Sal + Comm)

Select Ename, Sal, Comm, Sal + nvl(Comm, 0) as Tsal
 from Emp
 where Sal + nvl(Comm, 0) > 1500
 Order by Tsal;

4. Concatenation Operator (||)- pipes

It concatenates any 2 values @ columns

It is represented by 2 pipes (||)

SQL Server (+) is used

Select Ename || ' works as' || Job as details from Emp;

5. Logical Operator

is used for evaluating conditions, It is used only for comparing null & empty values.

OR → Binary → (A>10) OR (B<20)

Atleast one condition should satisfy / True

AND → Binary → (A>10) AND (B<20)

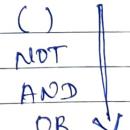
Both condition should satisfy / True

NOT → Unary → NOT (A>10) → Negation

Truth table

	AND	T	F
T	T	F	NOT(F) ⇒ T
F	F	F	NOT(T) ⇒ F
OR	T	F	
T	T	T	
F	T	F	

precedence →



AND - intersection

OR - Union

1. Display all the clerks from dept 20 & earning more than 2k

Select * from Emp

where deptno = 20 → Only once where is used
AND Job = 'Clerks'
AND Sal > 2000;

2. List all the employees whose name starts with 'A' or 'S'

Select * from Emp

where Ename Like 'A.%'
OR Ename Like 'S.%';

NOTE: NOT can be used like

NOT IN, NOT LIKE, NOT NULL, NOT BETWEEN, ~~IS NOT~~

Select * from Emp

where deptno in (10, 20);

deptno not in (10, 20);

Ename not like 'A.%';

Comm is not null;

Sal not between 2000 and 3000;

⇒ Select * from Emp

Where deptno = 20
AND Sal > 2000
OR Job = 'Clerks'

1 & 2 will be executed first and then 3

Select # from Emp

where deptno = 20
1] AND
AND (Sal > 2000
2] OR
OR Job = 'Clerks';
3])

2 & 3 will be executed first then 1

ASSIGNMENT

1. Display the deptname having atleast 2 Os (letter O)
(e.g., 10, 100...)
2. Display all the employees who are getting 2500 and excess salaries in department 20
3. Display all the managers working in 20 & 30 dept.
4. Display all the managers who don't have a reporting manager
5. Display all the employees who are getting some commission while their designation is neither manager nor analyst
6. Display all the analysts whose name doesn't ends with 's'
7. Display all the employees whose name is having letter 'E' as the last but one character.
8. Display all the employees who are getting some commission in dept 20 & 30
9. Display all the managers whose name doesn't start with A & S
10. Display all the employees whose earning salary not in the range of 2500 and 5000 in dept 10 & 20
11. List all the employees who are getting 2500 and excess salaries in dept 20
12. List all the employees whose name is having exactly 4 characters.
13. List all the employees whose job is having atleast 5 characters in dept 20;

13. List all the employees earning more than 2000 in
dept 30 with some commission. The output should be
arranged according to the date of joining

14. List all the salesmen from dept 30 who are earning
more than 100 as commission

15. Write a query which generates an output like
this

Scott works as manager in dept 20

Ford works as Analyst in dept 20

This should include only the employees from
10 & 20 depts.

16. List all the Clerks & Managers from dept 10 & 20
The output should not include any employees who
are getting salary less than 2000

17. List all the employees who are joined in the year
1981 excluding the one who are not belonging to
dept 10 and their name not starting with 'A' or 'S'

18. Display all the emp whose name is having letter 'S'.

19. Display the Emp whose name is having letter 'D' as
the third char

20. Display all Emp who join in the month of Sep

ANSWERS

1. Select dname from dept
where dname like '%O%/%'

2. Select ename, sal, deptno from Emp
where sal >= 2500 AND deptno = 20;

3. Select ename, job, deptno from Emp
where job = 'MANAGER' AND dept in (20, 30);
^{as mgrname}

4. Select ename, job from Emp
where mgr is null AND job = 'MANAGER';

5. Select ename, job, comm from Emp
where comm > 0 AND job not in ('MANAGER',
'ANALYST');

6. Select ename, job from Emp
where job = 'ANALYST' AND ename not like 'J%';

7. Select ename from Emp
where ename like '%E-%';

8. Select ename, comm, deptno from Emp
where comm > 0 AND deptno in (20, 30);

9. Select ename as mgrname, job from Emp
where job = 'MANAGER' AND
(ename not like 'A%') OR (ename not like 'S%');

01-JAN-1981
31-DEC-1981

10. Select * from Emp where deptno in (10, 20)
AND sal Not between 2500 AND 5000;

11. Select Ename from Emp
where Ename like '---';

(12) Select Ename from Emp
where length(Ename) = 4;

13. Select * from Emp
where deptno = 20
AND Job like '----%';

14. Select * from Emp
where deptno = 30
AND sal > 2000
AND comm > 0
Order by hiredate;

15. Select ename || ' works as ' || Job || ' in dept' ||
deptno as Emp-details from Emp
where deptno in (10, 20);

16. Select * from Emp
where Job in ('CLERK', 'MANAGER')
AND deptno in (10, 20)
AND Sal > 2000;

(17) Select * from Emp
where hiredate like '1/81' → (between
AND deptno = 10
AND (Ename not like 'A%') OR Ename not like 'S%');

(18) Select Ename from Emp
where Ename is like '1-S%'

(19) Select Ename from Emp
where Ename like '---O-%'

(20) Select * from Emp
where hiredate like '1-Sep-%'

SET Operators

If $A = \{2, 3, 4\}$
 $B = \{1, 2, 5\}$

$$A \cup B = \{1, 2, 3, 4, 5\}$$

$$A \text{ Uall } B = \{2, 3, 4, 1, 2, 5\}$$

$$A \cap B = \{2\}$$

$$A - B = \{3, 4\} \rightarrow \text{Elements only in } A \text{ but not in } B$$

$$B - A = \{1, 5\}$$

$$(A - B) \neq (B - A)$$

List of Set Operators

1. Union
2. Union All
3. Intersect
4. Minus

Set Operators helps to combine the data b/w any 2 queries
 (May be on the same @ diff tables)

UNION: Combines the data b/w any 2 queries,
 removes the duplicates and sorts the data in
 ascending order of the first column

UNION ALL: Combines the data b/w any 2 queries
 as it is. It is faster than UNION as
Union ALL will not do sorting and filtering.

Intersect: It returns the common value b/w the queries

MINUS: It returns the data which is exclusive to
 the first query (present in first query but not in
 second query)

Select * from Emp where Empno in (7839, 7788) 2
UNION

Select * from Emp where Empno in (7902, 7788); 1.

Select * from Emp where Empno in (7839, 7788) 2
UNION ALL

Select * from Emp where Empno in (7902, 7788); 2

Select deptno from Emp

INTERSECT

Select deptno from dept;

Select deptno from dept

MINUS

Select deptno from Emp;

Select Ename from Emp

UNION

Select dname from dept;

Select Ename from Emp

UNION

Select *|| dname from dept;

Select Ename, Empno from Emp
UNION

Select dname, null from dept;

Restrictions:

1. The no of columns b/w the queries must be same

Select Ename, Empno, Sal from Emp
UNION

Select dname, deptno, null from dept;

* This problem can be overcome by adding a dummy column (Null Column)

To distinct
and Ename

Select ...

UNION

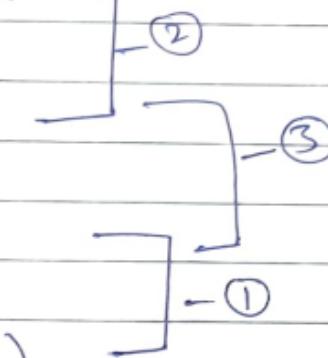
Select ...

MINUS

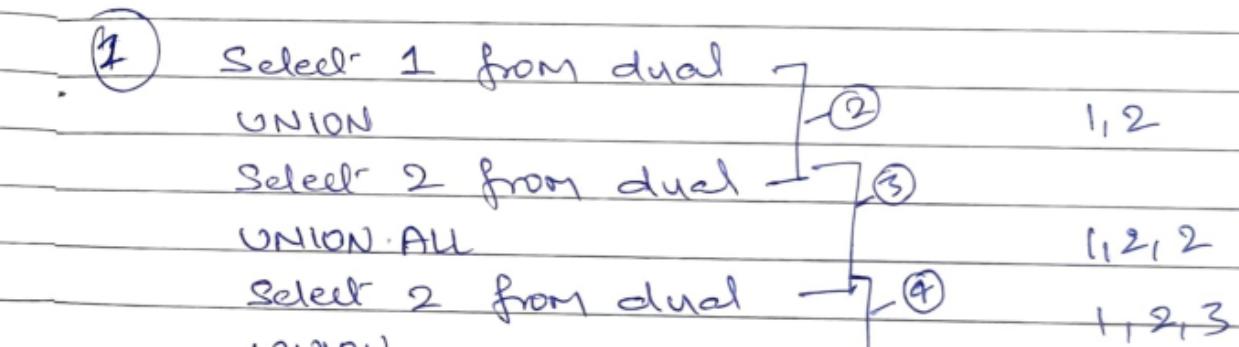
(Select ...)

INTERSECT

Select)



ASSIGNMENT



③ Q/p like below

* Scott - Emp Name
* blake - Emp Name

clerk - Job
Salesman - Job

@ Sales - Dname
@ Operations - Dname

Answer:

Select '*' || Ename || '-' Emp Name' as details
from Emp
UNION ALL

Select '#' || Job || '- Job' from Emp
UNION ALL

Select '@' || dname || '- dname' from dept
/

④ Q/p like below

Data

Enames

SCOTT
BLAKE

Dname

Sales
clerk

Select '*' || rpad(ename,10,'') || '-' Emp Name' as
details from Emp

Select 'Enames' as DATA from dual
UNION ALL

Select Ename from Emp;

Select '---' from dual
UNION ALL

ANSWER:

(Select 'Enames' as data from dual
UNION ALL

Select '-' || Ename from Emp)

UNION ALL

(Select 'Dnames' from dual
UNION ALL

UNION ALL

Select '-' || dname from dept)

UNION ALL

(Select 'Jobs' from dual
UNION ALL

Select '-' || Job from Emp);

⑤ Difference between UNION & UNION ALL

⑥ Display common deptno b/w Emp & dept table

⑦ Display the deptno which is exclusive to dept table

Group Functions & Grouping

(3)

Group functions are also called as aggregate functions

Group Function / Aggregate function

MAX - Returns the highest value

MIN - Lowest value

Avg - Average value

SUM - Total Value

COUNT - No of records / value

VARIANCE - Variance

STD DEV - Standard deviation

Select SUM(SAL), MAX(SAL), MIN(SAL), AVG(SAL)
from EMP where DEPTNO = 20;

Select COUNT(GMPCNO), COUNT(COMM), COUNT(*)
from EMP;

Select * from EMP;

COUNT(GMPCNO) - no of records not null in EMPNO in Column

COUNT(*) - no of records in table

Select COUNT(*) from EMP
where Job = 'CLERK' and DEPTNO = 20;

It displays no of clerks in deptno 20

Assignment

- What is the highest salary among Manager & Clerks
- How many employees are getting commission
- How many employees are not getting commission
- What is the Avg. Salary for the people who are joined in 1981 excluding the December month
- Display highest salary for all Salesman
- What is the Avg. Salary of all the managers from Dept- 20 and 30, who are earning less than 2500
- List the highest & lowest commission of all the salesman in Dept- 30, who are having reporting managers.
- Display the no of Salesman earning more than 1000 in Dept- 30
- Display the highest & lowest salary of all the managers.
- How many employees are having their names ending with 'es'

ANSWERS

① Select MAX(SAL) from EMP
where Job in ('MANAGER', 'CLERK');

② Select Job, MAX(SAL) from EMP
where Job in ('MANAGER', 'CLERK')
Group by Job;

③ Select COUNT(COMM) from EMP;

④ Select COUNT(*) from EMP
where COMM is not null @) COMM > 0;

Group Functions & Grouping

(3)

Group functions are also called as aggregate functions

Group Function / Aggregate function

MAX - Returns the highest value

MIN - Lowest value

Avg - Average value

Sum - Total Value

Count - No of records / value

Variance - Variance

STD DEV - Standard deviation

Select Sum(sal), Max(sal), Min(sal), Avg(sal)
from Emp where deptno=20;

Select Count(emph), Count(comm), Count(*)
from Emp;

Select * from Emp;

Count(emph) - no of records not null in Empno in column
Count(*) - no of records in table

Select Count(*) from Emp
where Job = 'CLERK' and deptno=20;

It displays no of clerks in deptno 20

Assignment

- What is the highest salary among Manager & Clerks
- How many employees are getting commission
- How many employees are not getting commission
- What is the Avg. Salary for the people who are joined in 1981 excluding the December month
- Display highest salary for all Salesmen
- What is the Avg. Salary of all the managers from dept 20 and 10, who are earning less than 2500
- List the highest & lowest commission of all the salesmen in dept 30, who are having reporting managers.
- Display the no of Salesmen earning more than 1000 in dept 30
- Display the highest & lowest salary of all the managers.
- How many employees are having their names ending with 'es'

ANSWERS

① Select Max(sal) from Emp
where Job in ('MANAGER', 'CLERK');

② Select Job, Max(sal) from Emp
where Job in ('MANAGER', 'CLERK')
Group by Job;

③ Select Count(comm) from Emp;

④ Select Count(*) from Emp
where comm is not null and comm > 0;