

Example :-

MS-Excel, dBase, Foxbase, Foxpro
Clipper, Advance Revelation, Database, Dataflex.
Quattro pro etc.

R RDBMS

→ Relational Database management system.

ex Oracle.

Oracle → OODBMS & RDBMS (Object oriented)

Oracle is known as (RDBMS + OODBMS)

Comparison between RDBMS & DBMS.

DBMS	RDBMS
a) Field b) Record c) File	a) column, attribute, method b) Row, Tuple, Entity, opportunity c) Table, Relation, Entity class matrix, Applet
1) Naming convention	
2) Relationship b/w two file is maintained programmatically	2) Relationship b/w 2 file specified at the time of table creating
3) More time needed for software development.	3) less programming.

DBMS

- 1) In DBMS we have high network traffic.
 - 2) Processing of client machine.
 - 3) Slow and expensive
 - 4) Client server architecture is not supported.
 - 5) File level locking.
 - 6) Not suitable for multiuse.
 - 7) Distributed database are not supported.
 - 8) No security of data
- (DBMS allow to access the data from outside the DBMS)
- 1) low network traffic.
 - 2) processing of server machine.
 - 3) Faster and cheaper
 - 4) Most RDBMS support client server architecture.
 - 5) Row level locking.
 - 6) suitable for multi-use.
 - 7) Most of RDBMS support distributed database
e.g. banking software
 - 8) Multiple level of security

RDBMS

- a) Logging in security
e.g. Database Username/ password.
- b) Command level security
e.g. create table, create view
- c) object level security
e.g. Accessing table to other user.
(Security is inbuilt feature of RDBMS)

Various RDBMS available :-

* Informix

→ Fastest in terms of processing

Oracle (most popular) (work with all OS)

Sybase (Going down)

MS SQL Server (good RDBMS) (only work with Windows)

* Open source based RDBMS

Ingres

PostgreSQL

Unify

} (work only on Unix OS)

DB2 (from IBM)

CICS

TELON

IDMS

} Server has to be mainframe computer (supercomputer)

1 Single user RDBMS

MS Access

Paradox

Lotus SQL

* Free and open source (work on multiple OS's)

MySQL

(Google, YouTube, Facebook, Twitter, WordPress, Joomla)

~~Oracle~~ www.oracle.com

- Most popular
- Work with 11g OS's
- Product of Oracle Corporation (1977)
- #1 Software Company in the world
- #1 Database Software Company in the world
- 63% world paid RDBMS market for client server environment
- 85% of world paid RDBMS market for internet Database.
- Best Software development tools.

~~MySQL~~

- Founded in 1995 by Swedish company
- Combination of "My" the name of co-founder's daughter and "SQL"
- Sun Microsystems acquired MySQL in 2008
- In 2010 Oracle Corporation acquired Sun Microsystems
- Part of widely LAMP software stack and other AMP stack.

L - Linux	W - Windows	M - Mac OS
A - Apache	A	A
M - MySQL	M	M
P - Python, PHP, Perl	P	S

→ even Rlicher uses MySQL
→ 42% of world open-source free
RDBMS market

\$ BEST SOFTWARE DEVELOPER TOOLS FOR Oracle 11g.

SQL

SQL * plus.

PL / SQL.

Oracle Forms.

Oracle Export.

Oracle menus.

Oracle graphics

IDS

EXP

IMP

DEM

SQL * Loader.

Oracle CASE

Oracle Financials

Oracle manufacturing

Oracle - HRMS

Oracle CRM

Oracle Application

personal oracle

Oracle Lite

etc.

SQL (Structured Query language)

- Commonly pronounced as "sequel"
- Create, Drop, Alter table, Insert, Update, Delete
Grant, Revoke, Select
- Conform to ANSI Standards * 2013 std.
e.g. 1 character = 1 Byte of storage
- Conform to ANSI ISO std (quality assurance)
- Common for all RDBMS hence also known
as standard query language.
- Initially founded by IBM (1975-1977)
Source code (80% C & C++ / 20% Assembly)
- Now controlled by ANSI
(hence common for all language)
- The entire source code of JAVA programming →
SQL is 100% JAVA programming.

product from Oracle corporation

SQL * PLUS

- Extension to SQL
- Used to remove the limitation of SQL
(extra 54 command)
- ex describe, etc.
- Used to setting the oracle environment
- ex linesize, pagesize, etc.
- oracle client software
- interface with database for running SQL
+ SQL * command.

PL/SQL (procedural language SQL)

- programming language from Oracle
- used for database programming.
ex fax-scale, HE-scale etc.
- 4GL (support object oriented programming structure)
- PL/SQL is the #1 DB programming language in the world overall
- PL/SQL is the #1 ^ programming language in the world

Oracle Forms

- Used for creating data entry Scenrys.

Oracle Report

- Reportwriter (GUI based) (windows based)
- Business Intelligence tool (BZ tool)

Oracle menus

- pull down, full screen, bag, popup, menus
toolbars, etc.

Oracle Graphics

- pie-charts, bar graphs, x-y graphs etc.

IDS

(Integrated Development suite)

- Oracle Forms + Reports + menus + Graphics
- Front end software from oracle
- Job of Front end software is input & output.

EXP

- Export
- Used for taking backup ~~of~~ ~~to~~ table

IMP

(Import)

- Used from Form restoring the backup

OEM

(Oracle Enterprise manager)

- Used for ~~for~~ DBA (Database administration)
- Create user, assign permission, performance monitoring, performance tuning etc.

SQL * Loader

- Use to load data from other RDBMS into oracle database

Oracle CASE

(Computer Aided Software Engineering)

- Case tool
- Used for RAD (Rapid application development)

Oracle Financials

- ERP Software (Enterprisewide Resource planning)
- Readymade software for financial appl'n

e.g. Citibank, Deolite, HSking & Sells,
Morgan Stanley, BSE (Bombay Stock Exchange)

Oracle Manufacturing

- ERP Software (Enterprisewide Resource planning)
- Readymade software for manufacturing industry

ex Godrej, L&T, Siemens, Whirlpool,
Videocon, VIP Luggage

- #1 SAP (40%)
- #2 Oracle Financial (30%)
- #3 J. D. Edwards (20%)

Oracle HRMS (Human Resource management system)

- ERP software
- Readymade software for (HRD)
(Human Resource Development)

e.g. ICICI BANK, Accenture,
Cognizant, capgemini, pepsi, etc.

#1 peoplesoft.

Oracle CRM

(Customer Relationship management)

→ ERP Software (Enterprise wide Resource planning)

→ Readymade software of CRM

ex Call center of IBM, ICICI Bank, LG, Samsung

#1 Siebens system

Oracle application

→ Oracle Financials + Manufacturing + HRMS + CRM

→ complete ERP solution

personal oracle

→ Single user version

→ available on DOS & Windows

Oracle Lite

→ Oracle for mobile phone

→ Available on android, ios, BB10, Symbian
Windows ME

Oracle Edition

→ All edition are free for learning, development, testing.

1) Lite edition

- mobile edition
- Free for end user
- Available on android, ios, windows ME
BB10, Symbian
- Single user version

2) personal edition

- Desktop / laptop edition
- Single user version
- Free end user
- available on DOS & Windows

3) Express edition

- Multiuser version
- Available on Windows, Linux / & Unix
- Free for end user
- DB size $\leq 10 \text{ GB}$.

4) Standard Edition

- Multiuser version
- Available on 113 OS
- No limit on DB size
- Rs 2-5 lakh (approximately)
- Advance features such as table partitioning clustering etc not supported.

5) Enterprise edition

- Complete Oracle
- available on 113 OS
- ~40 lakh price
- Advance features such as table partitioning clustering are supported

6) Version

V1, 2, 3, 4

DBMS

v5, 6

partial RDBMS

v7 → 100% RDBMS

(all constraint, distributed database client server architecture introduced)

v8 ORDBMS

v8i, 9i → internet computing

v10g, 11g → grid computing

v12c → cloud computing (Software as a Service (SaaS) platform as a service (PaaS))

My SQL

- Free RDBMS
- No limit on DB size
 - ⇒ (Facebook, youtube, etc)

V5.7 (latest version)

- open source
- available on windows, linux, unix, ios
- multiuser
- source code of my SQL is c & c++

Oracle SQL

- Structured query language
- Commonly pronounced "sequel"
- Common for all RDBMS
- Conforms to ANSI and ISO standard
- Initially founded by IBM
- Now controlled by ANSI
- because SQL is controlled by ANSI it is common for all RDBMS

4. Sub-division of SQL

DDL, DML, DCL, DQL

DDL (Data definition language) (Create, Drop, Alter)

DML (Data manipulation language)
(Insert, update, delete)

DCL (Data control language)

DTL / TCL (Data transaction language)
 (Transaction control language)
 → commit, Rollback, Savepoint

DPL (Truncate)

DML (Merge, Upset)

Rules For Tablenames, Columnnames, Variablenames

- Max 30 characters
- A-Z, a-z, 0-9 allowed
- tablename has to begin with alphabet
- special characters \$, #, - allowed

ex EMP2016, EMP_2016

→ 134 reserved words not allowed (check oracle Documentation)

documentation → (<http://docs.oracle.com>)

Oracle Datatypes :-

char

Number

Date

Varchar2

Long

Raw

Long Raw

char

- allows any character
- max upto 2000 characters
- E.g. Roll-No

Number (allows numbers only)

→ Range $\pm 1.0 * 10^{-130}$ to $\pm 9.9 * 10^{125}$

→ Example - SAL, COMM, etc

Date

→ allows date

→ 1st Jan 4712 BC to 31st Dec 9999 AD

a. → 78 date formats

b. → dt1 - dt2 (we can subtract date)

return the number of days b/w
2 dates

c. → internally Date is stored as number

d. → 7 byte of storage

1st Jan 4712 BC → 1

2nd Jan 4712 BC → 2

3rd Jan 4712 BC → 3

21st Aug 2016 AD → 2476398

(no of days since 1st Jan 4712 BC)

e. → 5th Oct 1582 AD to

14th Oct 1582 AD (missing day)

~~4th Oct 1582 AD~~

~~15th Oct 1582 AD~~ because Julian calendar to Gregorian

F. → date and time stored together

ex 10 Mar 1990 12:45:30 AM/PM

$dt_1 - dt_2 \rightarrow$ Number of days, remainder hours, minutes, seconds betn. the two days

$dt_1 - dt_2 = 1.5 \rightarrow 1 \text{ day } 12 \text{ hrs}$

→ whenever you don't type the time
the default time is 12 am midnight (00:00:00)
→ Time is optional

ex Date . e.g. DOB, HIREDATE, etc.

Varchar

- Variable character
- allows any character
- Max upto 4000 characters

ename. char (20)

ex Amit -----
by occupy (20bit)

b) Searching & Retrieval
very Pos

c) Wastage of handish space

ex PANNO, etc

~~Pos~~ Fix space

Varchar (20) ename

ex Amit.
by size will be 4 bit only

Searching & Retrieval
will be very slow

d) & required less space.

ENAME, CITY

Long

- allows upto 2GB of character data (variable length)
- large amount of text
- e.g., Experience, Resume, etc., comments

Raw

- allows upto 2000 bytes of binary data
- e.g. → ICONS, BARCODES, FINGERPRINTS, SIGNATURE etc

Long Raw

- Allows upto 2GB of binary data
- e.g. PHOTOGRAPH, SOUND, MUSIC, VIDEOS
- long Raw is multimedia datatype

~~↳ Oracle~~

- No limit on number of rows & per Table
 - Max 1000 columns per table
 - Can have only one column per table
(either long or long Raw)
- ⇒ ~~↳~~ → always stored the long columns in separate table because it slows down the processing speed for other columns

EMPNO EMP

EMPNO	ENAME	SAL	CITY	DOB
1	Amit	5000	Mumbai	10-Jan-1980
2	King	7000	Delhi	15-Feb-1978
3	LB	6000	OB	OB
4	Atul	8000	OB	OB

EmpNo Char(4)

Ename Varchar(25)

Say number (7,2)

7 → precision

2 → scale

12345.67

city Varchar(15)

Dob date

~~New table~~

Create table EMP

(

EmpNo Char(4),

Ename Varchar(25),

Say number (7,2),

city Varchar(15),

Dob date

);

; (; → End of command)

INSERT.

insert into emp

values ('1', 'Amit', 5000, 'Mumbai', '10-JAN-1980');

→ For Char, Varchar & date use ' ' (single quote)

' DD - MON - YYYY '

' DD - MON - YY ' → Default date format

US ' 21 - AUG - 16 ' → last two for current century
21 - AUG - 1969

accordance
insert into emp (empno, sal, ename, city, dob)

values ('2', 7000, 'Amit', 'Delhi', '15-FEB-1970');

insert into emp (empno, sal)
values ('3', 6000);

→ Null value is (ASCII value 0)

→ Special treatment given to null value
in database .

→ null value is independent of datatype

→ null value works with all datatype

is known as datatype-independent

→ null values occupies only 1 byte of
storage

→ If row is ending with null values those
columns will not occupy any space

→ Those columns that are likely to have
a large number of nulls should

preferably be at the end of the structure

(to conserve on HD space)

insert into emp
values ('4', 'Atul'); \leftarrow ERROR

insert into emp
values ('4', 'Atul', null, null, null);

SELECT

select * from emp;

NTFS - New Technology File System (64 bit)

PARTITIONS

NTFS 64 bit	OS WIN 2008 SERVER FILES	oracle DB LOG ORCLDB 450 GB	MySQL DB V5.7.14 DBNAME MYSQL57 486
	C: 200GB	D: E: .	: .

DB Server HD (1TB)

192.168.4.31

dace@dacea

Nitin/dace

Nitin/*****

BIN FOLDERS

oracle client\BIN
→ SQL PLUS

mysql client\BIN
→ MySQL workbench

Client machine
OS → Windows 10

shot - All APPS → Oracle - Oracle Client / SQL plus

Host String :-

server ip address, or machine name = 192.168.4.3

database name

protocol

port number

decora = { Shortcut created by MET)

{ Server machine = Host - (192.168.4.3)

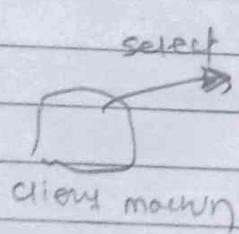
database_name / Service name = ORCL11g

Protocol = TCP (TCP or IPX)

Port number = 1521



Select * from emp;



* - metacharacter (all columns)

Server RAM (16GB)

→ READ
→ Complile
→ plan

DB Server (11g)

→ Execute

To restrict columns :-

Select empno, ename from emp;

Searching takes place here

Select deptno, ename, job, sal, hiredate from emp;

To Extract Rows :-

WHERE clause :-

{ Select * from emp
where depto = 10; }

WHERE clause is used for searching
WHERE clause is used for extracting the rows
from the dB Server Harddisk to RAM

{ select * from emp
where sal > 2000; }

Relational operators

1) $>$, \geq , $<$, \leq , \neq , $=$, \approx , \sim

order of operator precedence

$>$, \geq , $<$, \leq , \neq , \approx , \sim , $=$

{ Select * from emp
where sal $> 2000 \& 3000;$ }

Logical operators

- 1) NOT
 - 2) AND
 - 3) OR
- { precedence. }

$\left\{ \begin{array}{l} \text{select * from emp;} \\ \text{where deptno = 10 or sal > 2000 and sal < 3000} \end{array} \right\}$

$\left\{ \begin{array}{l} \text{select * from emp;} \\ \text{where (deptno = 10 or sal > 2000) and sal < 3000;} \end{array} \right\}$ will show all

$\left\{ \begin{array}{l} \text{select * from emp;} \\ \text{where job = 'MANAGER'} \end{array} \right\}$

$\left\{ \begin{array}{l} \text{select * from emp;} \\ \text{where } \cancel{\text{job is Manager}} \text{ job = 'Manager' or job = 'Clerk'} \end{array} \right\}$ o/p both rows

$\left\{ \begin{array}{l} \text{select * from emp} \\ \text{where job = 'manager' and 'clerk'} \end{array} \right\}$ will show nothing

$\left\{ \begin{array}{l} \text{select ename, sal, sal * 12 from emp;} \end{array} \right\}$

$\left\{ \begin{array}{l} \text{sal * 12} \Rightarrow \text{computed column (derived column)} \end{array} \right\}$

Arithmetic operators :-

precedence

1. ()

2. * * ex SAL * * 3 SAL³ / for square root
SAL * * (1/3)

3. /

4. *

5. +

6. -

** → does not work in Oracle SQL nor in MySQL SQL

** → work ~~is~~ in PL/SQL programming

{ select ename, say, say * 12 as "ANNUAL"
from emp; }
} optional

⇒ This concept is known as alias.

⇒ keyword 'as' is optional

⇒ as is ~~not~~ optional in Oracle & MySQL

⇒ as is ANSI SQL

⇒ If you want to case sensitive then
double quote is compulsory.

Select ename "EMP NAME",

say ~~"ANNUAL"~~, "salary",

say * 12 "ANNUAL",

~~say * 12 * 0.4 "HRA",~~

~~say * 12 * 0.2 "DA",~~

~~say * 12 + say * 12 * 0.4 + say * 12 * 0.2~~

"NEW EARNINGS" from emp;

→ where annual < 100000

← you can not use

alias in the WHERE clause

but we can write

where say * 12 < 100000

Select job from emp ;

Select distinct job from emp ;

⇒ distinct → to suppress the duplicate.

(So same name at a time)

internally works distinct

when setting distinct use sorting takes place
internally in the server RAM

clock
manager

Select job from emp ;

Select unique job from emp ;

some o/p

→ distinct will work in all RDBMS such as MySQL, MS SQLServer, etc,

→ unique will work in oracle.

Select distinct job, ename from emp ;

Select (distinct job), ename from emp ;
not allowed

{ EXIT } exit from server

Assign

{ 1, 2, 3, }

Oracle LG.

SQL command :-
extra in oracle :-

SQL> password

select * from tab; ← list of table

tab ← is an oracle created system table
stored the table name which you have created
2000 system table in oracle LG

A DATA DICTIONARY - is set of 2000
(Database catalog) system table in oracle LG

→ Set of system table in RDBMS

cmd
describe emp; → shows table structure type
(desc temp)

(1byte) SQL Buffer ← last SQL command is saved
buffer after saved

temporary text file on client machine in
BIN folder

→ Name of this file afredt.buf

/ → will execute what is saved in buffer

L1 Select
L2 *
L3 From
L4 Emp } — modify emp
 |

} — modify emp

PAGE No.		
DATE		

See for last command

~~END~~ LIST; ← will display contents of buffer

L1 → first line / L2 → second line / L3 → 3rd line

Eur; (R) → list and execute

~~For modify command → (* → enam)~~
modified first cmd select * from emp;
L2 →

(C/*/enam) (c-chang)

To add ~~list~~. Post o/p → enam

L2

a, say (a-append)

(/ → if we want to run edited command)

(O/P → emam, say)

Insert new line in command

L2

I, DeptNo

I where.

(,) is optional if we want to put in statement

want To delete 3rd line

L3

del ← delete for 3rd line

Ø

(editing command of sql plus)

SAVE <file name>

SAVE filename ;

SAVE ABCDE

→ ABCDE.SQL

→ By default saved in BIN folder.

SAVE c:\sameer\abcde ← saving for other folder

if want run this file after the crnp ↴

@ <filename>

@ abc

@ c:\sameer\abcde

SAVE abcde Eplace , * for overwriting file.

GET <Command name> ← saved command
← from buffer file

define_editor = notepad

define_editor = vi

? To open in
editor file

ed <filename>

ed → editor

ed ABCDE ;

← directly open this file
in editor

ed {
 | demon forward slash
 | can use multiple statement }

← will open the buffer file in notepad.

ED ← open command if better file in notepad

- point
- 1) do not remove ' '
 - 2) do not put ;
 - 3) can't use multiple statement
has to be SQL statement
- } buffer editor

~~spool <filename>~~

→ stored on client machine
in BIN folder

→ by default abc.lst extension

ex Spool abc

Spool : \ somees\abc

SQL > spool abc
SQL > select -----;
SQL > select -----;
SQL) select off
SQL > Select abc.lst ← stop the spool

{ O/P → O/P will show all file by
also send to the Bin file folder
some in (.LST) extension.

SQL > spool abc apped;

To add existing file

die → get list of file

SQL > host < os command >

for others than SQL command

→ 1) host die ; - will get list of file

2) host dir * .sql (p)

3) host ls

SQL

select deptno, job, ename, sal, hiredate
from emp;

→ ~~order~~ in the database, rows of a table
not stored sequentially in the database,
row of a table are scattered all
over the DB server HD (fragmentation
takes place)

→ When you select from a table, the order
of rows in the output depends on the row
row address

SQL (order by clause) ← end of slide

select ~~spets~~ deptno, job, ename, sal,
hiredate ~~order~~ from emp order by
ename;

→ By default it is ascending order

asc (by default)

* order by ename desc + for descending

~~Q~~

Select deptno, job, ename, sal, hiredate
from emp order by deptno, job;
(First it will sort dept no then job)

* order by lastname, firstname;

~~*~~ No upper limit on number of columns
in order by clause

* order by deptno desc, job;

* order by deptno desc, job desc;

* order by hiredate;

* Select deptno, job, ename, sal, hiredate
from emp

In this deptno = 10
order by ename;

- When clause is specified before the order by clause
- order by clause is the last clause in select statement

- * Select ename, sal * 12 from emp ;
- * Select ename, sal * 12 from emp
order by sal * 12 ;
- * Select ename, sal * 12 ANNUAL from emp
order by ANNUAL ;
- * Select ename, sal * 12 "Annual salary" from emp
order by 2 ; ← column number 2
- * Select ename, job from emp
order by sal ;

EMP

EMPNO	ENAME	SAL	CITY	DEPTNO
1	A.DAMS	1000	Bombay	10
2	BLAKE	2000	delhi	10
3	ALLEN	2500	Bombay	20
4	KING	3000	delhi	30
5	FORD	4000	Bombay	40

* Select * from emp
where ename > 'A' and ename < 'B';

✓ ✓ ✓ ✓
ADAMS > A - 6 8 6 8 8 .
✓ ADAMS B 6 6 6 f
1 2 1 = 1

BLANK - padded comparison semantics :-

- When you compare 2 strings of different lengths, the shorter of the 2 string is temporarily added with blank space on RHS such that their length are equal.
- Then the RDBMS will do the comparison of two strings characters by character based on ASCII value.

* Select * from emp
where ename >= 'A' and ename < 'B';

* SQL special operators (like) %, ^, any, IN

Select * from emp
where ename like 'A %';

% any character and any number of characters

% → wildcards

% → wildcards used for pattern matching.

* where ename like 'A %' or ename like '% a';

Special operators

- * select * from emp
 where ename like '~~A%~~', '%A';
 work only for Varchar
 ⇒ will get ename ending with A
- * select * from emp
 where ename like '%A%';
 o/p → name start & ending with A
- * select * from emp
 where ename not like 'A%';
 o/p → name not starting with A
- * Select * from emp
 where ename like '__A %';
 (underscor) → wild card o/p → names having 3rd letter is A
- * select * from emp
 where ename like '____';
 o/p → any four(4) letter name
- * select * from emp
 where sal >= 2000 and sal <= 3000;
 o/p → sal = 2000, 2500, 3000
 sal → between 2000 & 3000

Fuzzy
 Searching
 ex = google.

* Select * from emp
where say between 2000 and 3000;

executing →

1) Read & compile s/plo.
4) execute

Between → is a special operator

→ stored in database in compile form
the RDBMS directly plans & execute

→ select * from emp
where say not Between 2000 and 3000;
O/P → not betw 2000 & 3000

→ between operator work with all datatype

* Select * from emp
where hiredate ~~betw~~ between '01-JAN-15'
and '31-DEC-15';

* Select * From emp
where ~~hiredate~~ ename >='A' and ename <='B';

O/P → will name start with F but not FORD

ADMAS > A ⚡⚡⚡ {

A D Adams < B ⚡⚡⚡ { 1st condn

BLAKE > A ⚡⚡⚡ {

✓ BLAKE > ✓ B ⚡⚡⚡ { 2nd condn
greater than

→ * Select * from emp
where sal between 2000 and 3000 and
ename like 'A%' ;
O/P → Sal betn 2000 & 3000 and name
start with 'A'

→ * Select * from emp
where deptno = 10 or deptno = 20 or
deptno = 30 ;

* Select * from emp
where deptno = any (10, 20, 30) ;

⇒ any is special operator
→ It work same as OR operator.

* Select * from emp
where deptno in (10, 20, 30)

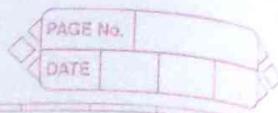
→ in also work as logical OR
→ any is better

* Select * from emp
where deptno not in (10, 20, 30)

* Select * from emp
where deptno != any (10, 20, 30)

* Select * from emp
where deptno > any (10, 20, 30)

Any → logical OR
IN → logical OR



* where deptno > any (10, 20, 30);

* where deptno >= any (10, 20, 30);

* where deptno <= any (10, 20, 30);



Select * from emp

where deptno = some (10, 20, 30);

→ Some is same as any operator avoid using some, use any some i

→ SOME is outdated, ANY is important over SOME

DDL → Create

DML → Insert, Delete, update

DQL → Select

UPDATE (DML command)

* UPDATE emp

Set sal = 10000

where empno = 1;

* update ~~not~~ emp

Set sal = sal + sal * 0.4

where empno = 1;

O/P → Sal increase by 40%

for multiple we OR / &

* Update emp

Set say = 10000, city = 'pune'
where empno = 1;

* Update emp

Set say = 10000
where empno = 1 or empno = 2;

* Update emp

Set say = 10000
where empno in (1,2);

* Update emp

Set say = 10,000
where city = 'Bomday';

* Update emp

Set say = 10,000, city = 'pune'
where city = 'Bomday';

* update emp

Set say = 10000;
o/p → will update every row

Delete,

* delete from emp
where empno = 1;

from → optional is oracle and compulsory
in everywhere

From → ANSI SQL

From → optional oracle

From → compulsory in MySQL & other RDBMS

* delete from emp;

where city = 'Bombay';

* delete from emp;

O/p → ALL the rows will deleted but table will not deleted

* Drop table emp;

O/p ⇒ table will be remove from dB

→ drop is a DDL command

→ you cannot use WHERE clause with drop table : ...

~~* Truncate~~

will delete all data from table

TRANSACTION PROCESSING

Commit

→ Commit will save all the DML changes since the last committed state

insert -

insert -

insert -

commit ;

(Trans 1)

update -

delete - - -

commit ;

T 2

insert - - -

update - - -

delete - - -

commit ;

T 3

→ when user issues a commit it is known as a Transaction

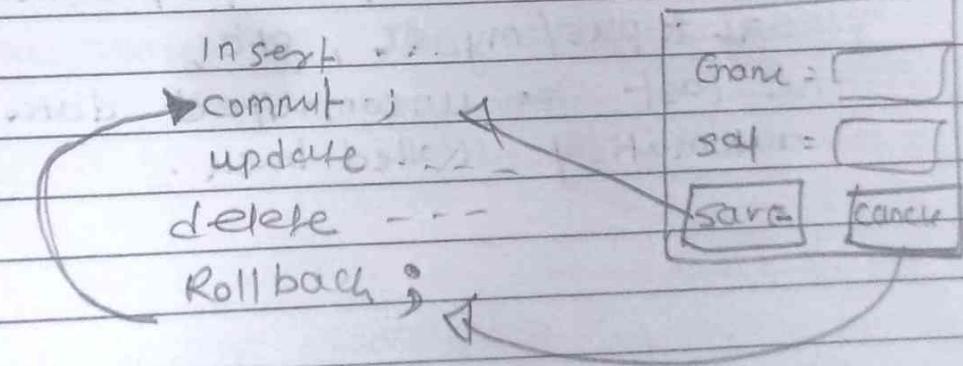
→ Commit will make the Transaction permanent

→ Related data constitute a transaction

→ Transaction is unit of work.

* Rollback - - -

→ Rollback will undo all DML changes since the last committed state



→ only the DML command are affected by Rollback and commit;

→ any DDL command automatically commits
not only the DDL command
commit itself but it will also
commit all DML changes prior to it

→ when you exit from SQL*PLUS,
it automatically commits
(even for my SQL)

commit ;

commit work; ← ANSI SQL

(work is optional in oracle &
my SQL)

Rollback ;

Roll back work ← ANSI SQL

(work is optional in oracle
and my SQL)

→ In case of power failure, network failure,
system failure, PC reboot,
windows close, improper exit from
SQL*PLUS/mysql, etc,
the last → uncommitted data is
automatically Rolled back.

Insert --

Insert ---

Insert ---

SAVE point ABC;

update --

update --

max 30 char

→ SAVE point PQR

delete --

delete --

Rollback TO RQL

→ Savepoint is a point within a transaction
(similar to bookmark)

→ you can Rollback to a savepoint
→ you can not commit to a savepoint
→ commit will save all DML changes since
the last committed state

→ when you Rollback or commit, the
intermediate savepoints are cleared
(they no longer exist)

→ If you want to use the savepoints again
you will have to issue them for a
fresh transaction.

→ you can only rollback sequential savepoint
→ you can not rollback between 2
intermediate savepoints

→ you can have two save point with the
same name (The latest savepoint
supercedes the previous one)

READ AND WRITE consistency.

→ When you ~~are~~ select from a table,
you can view only committed or other
changes

plus
changes made by you.

- when you update or delete a row
is automatically lock for user 3
- when you update or delete a row,
that row becomes READ-ONLY for
other users

* OPTIMISTIC LOCKING

- ROW LOCKING in RDBMS is automatically
- other user can select from that table
- They will view the old data before
before you changes
- other user can insert ~~as~~ Row into that table
- other user can UPDATE or DELETE
'other' rows from that table
- No other user UPDATE or DELETE
your locked row till you have
issued a Rollback or commit
- Locks are automatically released when
you Rollback or commit.

Doing JAVA or .Net & lock
the Rows
in advance

* PESSIMISTIC LOCKING -

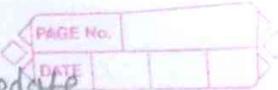
Rows in advance. Before
Issuing update or DELETE

Manual locking

For manual locking
will have to select statement with for
update clause

FIFO

S, C, 7, 8



for manually update

SQL > select * from emp for update.

SQL > select * from emp
where deptno = 10
for update;

locking

* locks automatically ~~are~~ released
when you commit or rollback

* Extra features on oracle :-

select * from emp
where deptno = 10
for update wait 60; ← 60 second.
It will wait for
60 sec

select * from emp
where deptno = 10
for noupdate;

* SQL Functions

SQL character function (common for all RDBMS)

EMP

FNAME

Arun

Tanvir

Sirun

LNAME

Purush

Azon

Kiran

Purush

Fname VARCHAR2(15)
Lname VARCHAR2(15)

Select * from emp;
Select Fname, Lname from emp;

* || → concatenation operator (works only in oracle)

* Select Fname || Lname from emp;

Output (Fname || Lname)

Arunpuran
tarunArun
Srirunkiran
Nutanpuran

* Select Fname || ' ' || Lname from emp;
blank space

OR Arun Puran

• Tarun Arun

* Select Fname || ' ', ' ' || Lname from emp;

OR

Arun, puran

Tarun, ATEUN

* Select 'ME.' || Fname || ' ' || Lname from emp;

'ME. "Fname ||", || lName (Alias)

O/P

ME. Arun, purn

ME. Tarun, Arun



concat (str1, str2)

→ working all RDBMS

e.g oracle, mysql, etc.

* Select concat (Fname, lName) from emp;

O/P

Arunpurn

Taran Arun

SIEUN KIEUN

NutanPurn

* select concat (concat (concat (fname, ' '), lName)
from emp;

O/P

Arun. purn

Taran Arun

SIEUN KIEUN

function within

function

max upto 255

leny for function within
function

* select upper (fname) from emp;

O/P

ARUN

TARUN

NUTAN

18/15
* update emp set fname = upper(fname);

It will update fname in uppercase.

* select * from emp where fname = 'ARUN'.

Case insensitive query :-

* select * from emp where upper(fname) = "ARUN".

* select lower(fname) from emp;

* update emp set fname = lower(fname);

Case -insensitive query :-

Select * from emp where lower(fname) = 'arun';

* initcap (work only in oracle)
Initial capital

Select initcap(ename) from emp;

O/P

Arun Puran
Tarun Arun

} - first letter in
capital & rest of
letters in small letter

* select lpad ('ename', 25, ' ') from emp;

↳ lpad → left pad

O/P

\$ \$ \$ \$ \$ Arun Puran
25

* Select lpad ('ename', 25, '#') from emp;

O/P

Arun puran
25

use of lpad

- 1) right - justification
- 2) cheque printing.

* select rpad ('ename', 25, ' ') from emp;

rpad - right pad

O/P

left
justified

Arun puran # # # # #
25

use

- 1) left justification for number deifer.
e.g (rpad (sal, 10, ' '))

spad 2nd use -

b) To convert VARCHAR2 to char datatype

c) cheque printing (use combination of lpad R spad)

* select spad(ename, 25, '*') from emp;

(pad (...) spad (say, ...)) ← e.g ↴

* select Ltrim(ename) from emp;

output

Arun puran

Tarun Arun

Siron KIUN

Nutan puran

↳ Arun & puran

↳ Tarun & Arun

↳ Siron & KIUN

↳ Nutan & puran

~~imp use~~

a) left justification

⇒ spad(Ltrim(ename), ...)

* Select Rtrim(ename) from emp;

R- Right trim → Remove the blank space
or Right hand side

Arun puran ↳

Tarun Arun ↳ ↳

~~use~~

a) convert char to varchar2

b) Right-justification of character data.

$\Rightarrow \text{lpad}(\dots, \text{trim}(\text{ename}), \dots)$

* select trim (ename) from emp;

\Rightarrow Remove the blank spaces from both side

(combination of ltrim & rtrim)

~~IMP~~

* select substr(ename, 3) from emp;

substr - substring.

3 \rightarrow starting position

O/P

un Puran

un Arun

un Kien

un Puran

substr ('New Mumbai', 5)

* select substr (ename, 3, 2) from emp;

O/P

un

yu

eu

ta

(1 2 3 4)

will get 3rd letter from
starting and ending with
4 letter

* select substr (ename, -3) from emp;

O/P \rightarrow will get last three letter in
ename

* Select substr(ename, -3, 2) from emp;

Output

ELI 9 3 2 1 0 -1 -2 -3 -4 -5
ELI . -3,2
ELI ()
 dm sm
 () ()

* Select replace(ename, 'un', 'xy') from emp

un → xy

O/P

Axy	Paxy
Taxy	Axy
Saxy	Kaxy

* Select replace(ename, 'un', 'xyz') from emp;

un → xyz

* Select replace(ename, 'un', x) from emp;

~~5x~~

1) Correction 2) Encoding/decoding

3) 1 → one

2 → TWO 3 → Three

spell out numbers

1) conversion \$ → RS

2) Encryption / Decryption

~~Translate~~ → Not supported in MySQL
 (works in oracle)

* select translate (ename, 'un', 'xy') from emp;

u → x n → y

~~op~~

Axxy	Pxxy
Taxxy	Axy
Sixxy	Wixxy
nxtay	Pxxy

* select translate (ename, 'un', 'xyz') from emp;

u → x
 n → y
 → z

* select translate (ename, 'un', 'x') from emp;

u → x
 u →

* select instr(ename, 'un') from emp;

→ is a character function but it
 return starting position of string

~~op~~

3
 4
 4
 20

④ (if does not found)

Aoun puran
 torun arrun

→ only 20 first observe

`INSTR` → used to check if one string exist in another one

Extra in Oracle :-

Select `INSTR(ename, 'un', 4)` from emp;

4 - . starting position

Output

9	Arun & Puran
4	Tarun & Arun
4	Sirun & Rishu

oracle

* select `INSTR(ename, 'un', 4, 2)` from emp

0	
9	
10	
0	

(only in oracle)

oracle

* select `INSTR(ename, 'un', -4)` from emp;

oracle (Oracle) op → starting from -4 position

* select `INSTR(ename, 'un', -4, 2)` from emp;

(Oracle) starting from 2nd of -4 pos

====

Comments

* `select`

{ L=M/P
Arun
Banerjee
charlie }

New table. For new command

* select length (cname) from emp;

Q1P 4
9
7

* select length (ltrim(cname)) from emp;
← For char column

* select ascii (cname) from emp;

Q1P 65 66 67 { Arun A-65
Banerjee B-66

* select ascii (substr(cname, 2)) from emp;

Q1P Ascii value for 2nd letter

* select ascii ('z') from emp;

Q1P .122 122 122 { 2 = 122 ASCII value
- ; here only 3 rows
∴ It will conn 3 time
→ If I

* select ~~distinct~~ ascii ('z')

* Select distinct ascii (ename) from emp;

Q122

* Select ascii ('z') from dual;

Q122

dual is a system table

dual contains only 1 row & 1 column

dual is a dummy table.

Select 3*12 from dual

Select 'welcome to MET' from dual

Select ascii ('A') from dual;

* Select lpad('CDAC', 10, ' ') from dual;

~~oracle~~

* Select char(65) from dual;

Q12

A



work in oracle in my SQL

mysql

* Select char(65 using utf8) from dual;

Q12

A

37

where **utf8** is the character set for which
it should display the character
return the 65 ASCII value in
english

* `Select * From emp where ename = 'Arun';`

* `Select * from emp where soundex(ename)
= soundex ('Arun');`

O/P =P sound Axile Arun

{ Arun
Arunn }
Arunn }

~~SQL Number Function~~

EMP

SALF

1234.567

1875.019

1348.156

1752.451

say number(7,3)

* `select round(sal) from emp;`

O/P 1235 / 1875 / 1348 / 1752

* `select round(sal,1) from emp`

O/P

1234.6

1875

1348.2

1752.5

} - If it is greater than 0.5 then it will add 1 in previous

* Select round (say, 2) from emp;

O/P

1234.57

1875.02

1348.16

1752.46



* Select round (say, -2) from emp;

O/P

1200

1800

1300

1800

* Select round (say, -3) from emp;

1000

2000

1000

2000

all the process taking place
in server

Oracle

* Select trunc (say) from emp;

trunc → truncut

O/P

will cuts off the decimal

1234

1875

1348

1752

Oracle

Select trunc (say. 1) from emp

O/P

1234.5

1875

1348.2

1752.5

* select trunc(sal, -2) from emp;

O/P

1200

1800

1300

1700

only in MySQL

* select truncate(sal, 0) from emp;

can not say sal only

O/P 1234 / 1348
 1875 / 1752

* select truncate(sal, 1) from emp;

* select truncate(sal, -2) from emp;

won't work for all

ceiling

* select ceil(sal) from emp;

1235

1875

1349

1753

used → bill payment

* select floor(sal) from emp;

(sal, -1) (sal, 2)

1234

1875

1348

1752

not possible

~~For oracle~~
MySQL

FLOOR will goes to lower value

from dual

* Select trunc(3.6), FLOOR(3.6), trunc(-3.6), FLOOR(-3.6)

3 3 3 -3

-4

~~For MySQL~~

* Select truncate(3.6), FLOOR(3.6), truncate(-3.6), FLOOR(-3.6)

3 3 -3 -4

due;

room

* Select sign(-15) from dual;

sign -1 → it return the sign of number

if n > 0 then
 Return + ;

IF n < 0 then return -1 ;

if n = 0 then
 Return 0 ;

~~Sign~~
Sign (balance)

Sign (fax)

Sign (marks)

Sign (bloodgroup)

Sign (temperature)

$x \rightarrow 10$

$y \rightarrow 20$

sign ($x-y$)

sign ($s_p - c_p$)

use of greater than
or smaller than

* Select mod (9,5) from dual;

which will integer or float

→ it return the remainder 9/5

op → 4

* select mod (8.22, 2.2) from dual

op 1.62

* select sqrt (81) from dual;

only with +ve numbers

op 9

If -ve then enter null

* select power (10, 3) from dual;

op 10^3

1000

* select power (10, 1/3) from dual;

op ~~sqrt~~ root

op 2.1544345

* select abs (-10) from dual;

⇒ 10 always return positive number
← setur absolute value

$\sin(x)$
 $\cos(x)$
 $\tan(x)$

$x \rightarrow$ has to be in radians

Extra in Oracle (will not work in MySQL)

{
sinh(x)
cosh(x) $x \rightarrow$ has to be in radians
tanh(x)}

Common for all RDBMS

$$\log(n, m) \rightarrow \log(n, m) \rightarrow \log_n(m)$$
$$\ln(x) \rightarrow \log_e(x)$$

SQL Date Functions & Formats :- (DD-MM-YY)

ANSI - does not recognize date-type

Oracle 11g

EMP
Hiredate

15-OCT-15
31-DEC-15
15-JAN-16

- 1) 1st JAN 4712 BC to 31st DEC 9999 AD
- 2) DD-MON-YY / DD-MON-YYYY
- 3) Internally date is stored as a number
number of days since 1st JAN 4712 BC
that number is known as Julian date

4) $dt_1 - dt_2$

5) 5th Oct 1582 AD 14th Oct 1582 AD
(Julian to Gregorian)

6) 2 bytes of storage.

date and time stored together.

default time 12 am midnight

* select sysdate from dual;

op will return system date & time

24 - AUG - 16

→ Returns server date & time

→ Returns the db server date and time.

* select sysdate + 1 from dual;

will return tomorrow date

op

25 - AUG - 16

* select sysdate - 1 from dual;

will return yesterdays date

op

23 - AUG - 16

* select sysdate - hiredate from emp

→

278 / 216 / 152

~~#~~ No of day betn two days

op. 278.7587

216.7587

152.7587

}

- 7587 → is remain time
as a fraction of day.

* select round($\text{sysdate} - \text{hiredate}$) from emp;

Q1P 278

216
152

* select months_between($\text{sysdate}, \text{hiredate}$) from emp;
will give me no of months
between the two days

Q1P 18 / 72 / 14.

18.1239

72.5138

14.0742

}

Explanation is no of days and
(days, hour, min + sec)

* select months_between($\text{sysdate}, \text{hiredate}$) / 12 from emp;
will give no of years in between
two days

* select add-month($\text{hiredate}, 2$) from emp;
will add 2 month in hire date

Q1P

15 - DEC - 15

29 - FEB - 16

15 - MAR - 16

* select add-month($\text{hiredate}, -2$) from emp;
will subtract 2 month from
hire date.

* select add-month($\text{hiredate}, 12$) from emp;
will add 1 year
but do not add in 365 days
* it is not correct

only available in oracle & MySQL

* select last_day (chieldate) from emp;

o/p

will return last day of month

31 - OCT - 15

31 - DEC - 15

31 - JAN - 16

sequred in Attendance by
emi payments

* select next_day (sysdate, 'Thursday') from dual;

o/p

25 - AUG - 16

will return the date of
comming thursday

* select next_day (sysdate - 7, ' Thursday') from dual;

will return the date of previous
thursday

~~*~~

Oracle 11g SQL date & time format
(DD-MON-YY)

* select systdate from dual;

o/p

24 - AUG - 16

(DD/MM) (MM/DD)

* select to_char (sysdate, "DD/MM/YY") from dual;

to_char -> convert date in char
string

→ 24/08/16

* select to-chaz (sysdate, 'DD-MM-YY') from duq;

DD	1) British	/	:	French	← ④
MM	2) American	-	*	Italian	← ⑤
YY	3) German	.			

⑥ - we can use blank space

DD - MM - In lower case also

⑦ YYYY - 2016 / 1995

⑧ YY - 016 / 995 / - barcode

⑨ YY - 16 / 95

⑩ Y - 6 / 5

⑪ MONTH → AUGUST / OCTOBER

⑫ month - august / october

⑬ Month - August / October

⑭ MON - AUG — by default

⑮ mon - aug

⑯ Mon - Aug

⑰ DAY - WEDNESDAY

⑱ Day -

⑲ day -

Select to-chaz (sysdate, 'DAY') from duq;

⑳ DY -

㉑ dy

㉒ Dy

(23) YEAR - TWENTY SIXTEEN
 Year - Twenty Sixteen
 year - twenty sixteen

* select to_char (sysdate, 'DDTH') from dual;

O/P 24TH 21ST 22ND 23RD

~~ddth~~ - 24th ~~24th~~
 DDTH - 24TH

* select to_char (sysdate, 'DDSPTH') from dual;

~~dd~~ TWENTY FOURTH

DD - MM

DDTH - MMTH

DDSP - MMSP

DDSPTH - MMSPTH

* D → 1-7
 DD → 1-3 | } - will return the value
 DDD → 1-365
 W → 1-5 → return something, being 1, 2, 3

WW → week of year → 1-53

* select to_char (sysdate, 'J') from dual;

O/P some number
 Date of Julian

Select hiredate + 1 from emp;
 Select hiredate + 1 from emp;

but we can't ↴

Y select '10/10/2016' + 1 from emp;

→ Select to_date('10/01/2016', 'DD/MM/YYYY') + 1 from
duey.
OR LL - JAN - 16

⌘ select to_char(to_date('10/01/2016', 'DD/MM/YYYY'),
 'DD/MM/YYYY') from duey;
OR

LL - JAN - 2016

⌘ after session set nls_date_format = 'DD/MM/YYYY';
 for changing date format

only for current session
 nls → National language support

(commit 1) oracle corporation)

⇒ If we want to permanent this format then
 go to the SQL startup file. Name → global login

- This file automatically executed when you start SQL * plus on your system
- Add any SQL & SQL*plus commands here that you want to be executed when user starts SQL*plus located in

→ open with word pad → at the end of
file

~~alter session set nls_date_format = 'DD/MM/YY'~~

~~IMP~~

select to_date ('04-oct-1582', 'DD-MON-YY') + 1 from
 duey;
 →
 15-oct-1582

~~* Oracle 11g SQL time format~~

select to_char (sysdate, 'DD/MM/YYYY
 HH:MI') from duey;
 HH - hour
 MI - minute
 (HH:MI:SS)

for 24 hr format (HH24:MI:SS)

for AM or PM (HH:MI:SS AM)

AM	PM
a.m.	p.m.
A.M.	P.M.
A.M.	P.M.

* insert into emp
values (to_date('10-APR-2016 11:30 AM',
'DD-MON-YYYY HH:MI AM'));

*
~~system~~ sysdate + 1 →
sysdate + 1/24
sysdate + 1/(24 * 60)
sysdate + 1/(24 * 80 * 60)

* select to_char(sysdate, 'DD/MM/YYYY HH:MM:SS AM'
"Mumbai TIME", to_char
sysdate - 5.5/24, 'DD/MM/YYYY HH:MM:SS AM')
LONDON TIME" from dual;

~~MySQL~~

Source code of C++

IP address - 192.168.4.83

1 TB HD

MySQL client SW

MySQL workbench

3.3

OS
WIN
2008

DB mysql-enterprise edition

VSA

DB - MySQL 5.7

EMP

OS
WIN
2008

SQL 1-23 → Using oracle
Ques 1-7 → My SQL
Assn - 1-6



- ↳ clte - ente - run
- show database ; — open database
- use Dac ...
- select user from user ;
- use user_date ;

- show tables ; (To show all table)
- desc emp ; — (structure of table)

* update emp set sal = 10000 ;
delete from emp ;

(does not delete)

(it not work)

initially

* For that ^(edit) menu → preferences → SQL editor

← F10

safe update
(unchecked)
not

login again → Query - disconnect to server

* Auto commit transaction

query - Auto commit transaction unchecked

Query → New tab to current server

(multiple new tab to
current server)

drop table <tablename>

Show <.table>

MySQL datatype :-

char.

Number. (int float

Date.

* character datatype

char & varchar

char (

→ allows any character.

→ max 255 characters

→ fixed length

ex (ROLL_NO, EMPNO, PANNO, etc)

varchar

→ allows any character

→ max 65,535 characters

→ Variable length

e.g CENAME, ADDRESS, CITY, etc)

→ limit 64kb (1) ⇒ 65535

end of string

(we don't have
long in SQL)

we have MySQL

Text

Text column stored outside the table

Tinytext

→ max 255 characters)

Text → max 65,535 characters

mediumtext → max 16,777,215 characters (16 mb)

longtext → max 4,294,967,295 characters (4gb)

- ⇒ Text columns are stored outside the table
- used for columns that are required only for display purpose
- used for columns that are not used in searching.

Multimedia Datatype

Binary - (Fixed length binary string)
(max 255 byte of binary data)

VARBINARY -

- variable length binary string
- max 65,535 byte of binary data

e.g. BARCODE, FINGERPRINTS, SIGNATURE etc.

e.g. GRAPHS, MAP, etc.

⇒ Searching & Retrieval will be very fast

large object datatype (~~LOB~~ LOB)

B(LOB) (BLOB) binary large data type

Tinyblob ↳

- max 255 B of binary data
- stored outside of datatable
- stored away from the row

- MySQL maintains a LOCATOR from the table

→ not recommended if this column is used for
row of this column

Blob (max 65,535 B of binary data)

mediumblob (max 16,777,215 B of binary data)

longblob (max 4,294,967,295 B of binary data) (4GB)

* enum →

→ enumerating

→ each column's value may be assigned one enumerating member

Ex → can define max 65,535 values

(create table)

tshirt-size enum ('xs', 's', 'm', 'l', 'xl', 'xxl'),

}

insert into tablename ---- values(1, '2016-08-25', 'M')

→ example use in radiobutton -----);

Set (use & choose more than 1 option)
→ (max of distinct numbers).

es Ice cream

- vanilla
 - chocolate
 - strawberry

IceCreamScope : set ('v', 'c', 's')

`insert into <tablename> values ('r', 'c');`

\Rightarrow values (\perp , '2016-08-25', $\underbrace{('r', 'c')})$;
Set datatype

Number datatype

Int / float

Signed or unsigned

signed - allows -ve values

unsigned - allows only positive value

Tinyint = (-128 to +127) or (0 to 255)

~~eg~~ - AGE

- small int

$\rightarrow -32,768$ to 32767 signed
 0 to $65,535$ unsigned

Medium int

- 16 million
- (-83 88608 to 83 88607)
- 0 to 15777215

Int

- 4 billion
- (-2147483648 to 2147483647)
- (0 to 4294967295)

Bigint

- (18 quintillion)

Signed - (-9223372036854775808 to
~~9223372036854775807~~)

unsigned

~~0 to 18446744073709551015~~



Float (single precision) (Signed only)

- (-3.402823466E+38 to
 $-1.175494351E-38)$

and ($1.175494351E-38$ to
 $3.402823466E+38)$

double - (double precision)

- (-1.7976931348623157E + 308 to
- 2.2250738585072014E - 308) and 0 and
(2.2250738585072014E - 308 to
1.7976931348623157E + 308)

Fixed-point ~~datatype~~ (exact value)

Decimal

- stores double as character string
- e.g. "653.7")
- max number of digit is 65
- used when it is important to preserve exact precision, e.g. monetary data,

Boolean datatype

- stored true & false
- equivalent to 1 & 0 respectively

e.g.
Munital status boolen

~~if~~ insert into --- (true);
insert into --- (false);
insert into --- (1);
insert into --- (0);

Date :-

1) Date / 2) Time / 3) Datetime / 4) Year

Date

$\rightarrow ('YYYY-MM-DD')$ ($'1000-01-01$ to
 $'9999-12-31'$)

- year values in the range 70-99 are converted
 $(1970-1999)$
- Range 00-69 are converted to 2000-2069

Time ->

$('hh:mi:ss')$ or $('hhh:mm:ss')$

range $('838:59:59' : to '838:59:59)$

Datetime

$('YYYY-MM-DD hh:mi:ss')$
 $\cdot ('1000-01-01 00:00:00' : to
 9999-32-12 23:59:59')$

Year

$('YYYY')$ 1901 to 2455

MySQL

Largest databases in the world

AMAZON → 100 terabyte
daily data

→ ID ~~SQL~~ MySQL 4096 columns per table
provided row size $\leq 65,535$ bytes

→ No upper limit on number of rows per table provided

→ table size \leq 64 Terabyte

* function

Character, Number, Date function ← some of oracle
list function

SQL List function

- Independent of datatype

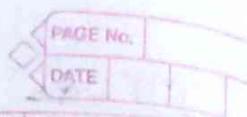
EMP

ENAME	SAL	COMM
A	5000	500
B	6000	
C		700

→ any comparison done with null, returns null only.

→ pessimistic querying:-

Oracle MySQL
nvl ifnull
null value ← →



is null is → special operator

* select * from emp where comm != null;

* select * from emp where is NOT null;

* select sal + comm from emp;

→ any operation done with null it sets null
~~nvl - oracle~~

→ any comparison done with null it sets null

* select sal + nvl(comm, 0) from emp;

~~off~~ 5500
6000

* select ~~set~~ nvl(sal, 0) + nvl(comm, 0) from

we can take

nvl(comm, 0)

nvl(comm, 100)

nvl(city, 'mumbai')

~~oracle~~ nvl → work with all datatype :

nvl → is not supported in mySQL
work

* select sal + ifnull(comm, 0) from emp;

EMP

SAL

1000

2000

3000

4000

5000

greatest

* select greatest (sal, 3000) from emp;

O/P

3000

o/p → set lower limit on something

3000

bonus

3000

4000

5000

* select greatest (val1, val2, val3 ... val255).

→ programming

max-num = greatest (x, y, z);

greatest (num1, num2, num3)

greatest ('ste1', 'ste2', 'ste3', 'ste4').

greatest ('dt1', 'dt2', 'dt3')

greatest

work with all datatype

* Select least (sal, 3000) from sal;

O/P

1000

2000

3000

3000

3000

} - use to set upper-limit
on something
e.g. cashback on 2000

- least (num1, num2, num3)

least ('st1', 'st2', 'st3')

least ('dt1', 'dt2', 'dt3')

work on all data type

EMP

SAL	DEPTNO
1000	10
2000	10
3000	20
4000	30
5000	40

oracle

decode → not supported by MySQL
(work only in oracle)

* Select decode (deptno, 10, 'Training', 20,
'Exports', 30, 'Sales', 'Others')
from emp

O/P

IF dept = 10	return	(O/P)
ZF dept = 20	—	Training
dept -		Training
		Export
		Sales
		Others

\rightarrow IF we don't specify other don't return
 $(dept = \text{something})$: it will return null.

2 Select decode (dept, 10, 'Ten', 20, 'Twenty',
 30 'Thirty', 40 'Fourty') from emp

O/P

Ten
 Twenty
 thirty
 Fourty

en purchase - p
 m - manager

 same functionality in MySQL using CASE expression

my SQL

```

Select CASE
    when dept_no=10 then 'Ten'
    when dept_no=20 then 'Twenty'
    when dept_no=30 then 'Thirty'
    else 'Other'
end
from emp;
    
```

IF deptno 10 = then HRA \rightarrow 40% annual
 IF deptno 20 = then HRA \rightarrow 30% annual
 IF deptno 30 = then HRA \rightarrow 25% annual
 else HRA 10% annual

~~\$~~
 select deptno , ename, sal * 12 "ANNUAL",
 case
 when deptno = 10 then . sal * 12 * 0.4
 when deptno = 20 then sal * 12 * 0.3
 when deptno = 30 then sal * 12 * 0.25
 else sal * 12 * 0.1
 end "HRA"
 from emp;

~~oracle~~

Other function .

- * Select usee from duay ; will return usee name
IN MySQL
- * select useec() from duay; will return useem

~~\$~~ SQL group function/ Aggregate function .

EMP (DB, Servy, HD)

EMPNO	ENAME	SAL	DEPTNO	JOG	MGR
1	Arun	8000	1	m	9
2	Ali	7000	1	c	1
3	Kiran	3000	1	c	1
4	Jack	9000	2	m	.
5	Thomas	8000	2	c	9

~~Not work in MySQL~~

PAGE NO.
DATE

* `Select decode (job, 'M', 'MANAGER', 'C', 'CLERK')`
from emp

~~oracle / MySQL~~

`select case`

`when job = 'M' then manager`

`when job = 'C' then CLERK`

`end`

`from emp`

Single-row function :-

Choe, Number, Date, List other Function

e.g. upper(ename), round(sal, 1), etc.

Multi-row function :-

Group Function

e.g. sum(sal)

* `Select sum(sal) from emp;`

O/P

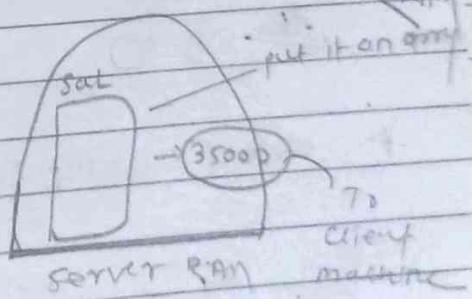
35000

internally - entire sal brought into

Server RAM

→ If anyone is null

→ Null value are not
counted by group function



Assume

last row sal is null :-

* `Select sum(sal) from emp;`

O/P 29000

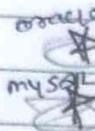
* select $\text{avg}(\text{sal})$ from emp;

Q/P

$$29000/4 \rightarrow 7250$$

} It will not count null rows

→ Assume 1 value is null



select $\text{avg}(\text{null}(\text{sal}, 0))$ from emp



select $\text{avg}(\text{ifnull}(\text{sal}, 0))$ from emp

Q/P

$$29000/5 \Rightarrow 5800$$

} It counts the null rows

} depend on U.S.E.R

* select $\text{max}(\text{sal})$ from emp;

→ Q/P → will return maximum of sal

Q/P

9000

* select $\text{min}(\text{sal})$ from emp;

3000

} ask to use whichever have to select

* select $\text{min}(\text{null}(\text{sal}, 0))$ from emp;

& select $\text{min}(\text{ifnull}(\text{sal}, 0))$ from emp;

Q/P ⇒ 0

* select $\text{count}(\text{sal})$ from emp;

Q/P 4

→ Returns count of number of rows where SAL is not having a null value

* Select count (*) from emp;

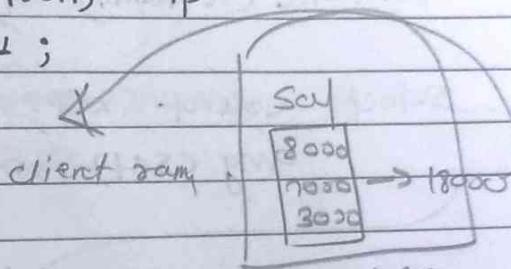
O/P 5

⇒ Returns count of total number of rows in the table

→ Assumption, last row say is 8000 :-

* Select sum (sal) from emp
where dept_no = 1;

→ 18000



⇒ where clause is used to select rows from DB server HD to server RAM

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

→ O/P 8000

8000

→ Count-QUERY (Counting the no of query hits) :-

* Select count (*) from emp
where sal > 7000;

O/P 3

* Select max(sal) / min(sal) from emp;

O/P 3

* select sum (sal) / count (*) from emp ;

sum (column)	}	number datatype
avg (column)		
min (column)	}	will work with all datatype min (hasedate), max
max (column)		
count (column)	}	- number datatype
count (*)		
std dev (column)	}	-
Variance (column)		

& select count (*), min (sal), max (sal), sum(sal),
avg (sal) from emp ;

summary report

OP

{	Count (*)	min(sal)	max(sal)	sum(sal)	}
	14	↓	↓	↓	
()					

(Select ename, sum(sal) from emp)

does not allow

error → This is not a single-group function
(Select ename, count(ename), sum(sal) from emp)
It will allows

→ You can not select a total aggregate column along with group function.

Select upper(ename), sum(sal) from emp;

→ It will not allow

→ You can not SELECT a single row function alongwith group function;

Select * from emp
where sal > avg(sal); } not allow

→ You can not use ~~other~~^{group} function in the where clause

Group by clause (used for grouping)

EMPNO EMP Table. (DB Server)

EMPNO	ENAME	SAL	DEPNO	JOB	MGR
1	Arun	8000	1	M	4
2	Ali	7000	1	C	1
3	Kiran	3000	1	C	1
4	Jack	9000	2	M	4
5	Thomay	8000	2	C	4

- Database schema by the name of `dbo` has been created
- `dbo`'s tables are stored in `dbo` schema
- `dbo.emp`

For Creating Schema

will have to login on DB

→ For accessing other user table

Select * from `dbo.emp`; serve to RAM



Sum (say) deptwise:

1	8000
1	9000
1	3000
2	9000
2	8000

Select deptno, sum(say) From emp
group by deptno;

~~Q1~~ Dept No sum(say)

1	18000
2	17000

calculation takes
place in server RAM

① Rows are retrieved from dB serve HD to
serve RAM

- 2 → Sorting will take place deptwise
- 3 → Grouping will done deptwise
- 1 → summation is done

1	8000
1	9000
1	3000
2	9000
2	8000

18000 17000

→ select clause → $\left\{ \begin{array}{l} \text{select deptno, sum(sal)} \\ \text{from emp} \end{array} \right.$
 → from clause → $\left\{ \begin{array}{l} \text{from emp} \\ \text{group by dept} \end{array} \right.$
 → group by clause → $\left\{ \begin{array}{l} \text{group by dept} \end{array} \right.$

① whichever column is present in select clause, has to be present in group by clause

→ $\left\{ \begin{array}{l} \text{will} \\ \text{error} \end{array} \right\}$
~~select * deptno, sum(sal) from emp;~~
 → ~~select count(deptno), sum(sal) from emp;~~

* → select deptno, sum(sal) from emp
 group by deptno;

→ whichever column is present in group by clause, it may or may not be present in select clause

* select sum(sal) from emp
 group by deptno

~~sum(sal)~~

18000

17000

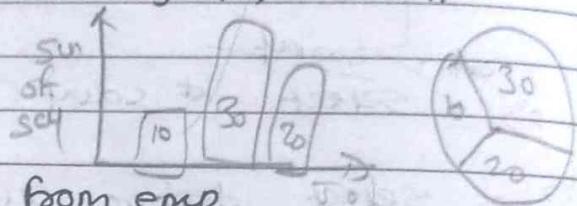
* select deptno, sum(sal) from emp group by deptno;

* select deptno, min(sal) from emp
 group by deptno

* select deptno, max(sal) from emp group by dept no;

* select deptno, count(*) from emp group by dept no;

 2D query => any select with GROUP BY clause we can plot a graph out of it



* select job, sum(sal) from emp group by job;

* select deptno, sum(sal) from emp
where by sal > 7000 group by deptno;

 where clause -> is specified before the group by

 select upto bottom of where before the rows in RAM

 where is used to Extract rows to form series HDD to Searce RAM

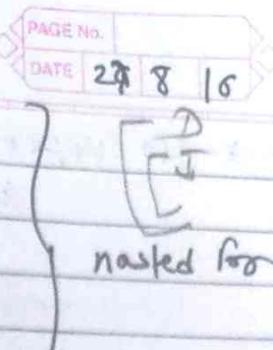


DeptNo	sum(sal)
1	8000
2	17000

* select deptno, job, sum(sal) from emp group by deptno, job;

op

deptno	Job	sum(sal)
1	C	10000
1	M	8000
2	C	8000
2	M	9000



If we want to write a program for this op will have to use nested for loop

* select job, deptno sum(sal) from emp
group by deptno, job, deptno;

op

C	1	1000
C	2	8000
M	1	8000
M	2	9000

⇒ The order of column in select clause and the order of ~~the~~ column in group by clause need not be the same

→ The order of column in select clause will determine the order of column in the output

-) The order of columns in group clause will determine the sort, grouping order, summing order

→ The order of columns in group clause will ~~not~~ determine the speed of execution

→ No upper limit on the number of columns in group by clause



Select -----

group by country, state, district, city;

order must be same
will affect the speed of execution



1 column in group by clause give me 2D query

2 columns in group by clause give me 3D query

3 columns in group by clause give me 4D query

etc.

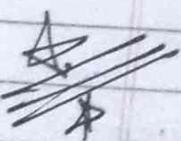
Known as multidimensional query

(also name of spatial queries)

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

may or may
not be available in group by
Select

but deptno is now available in group clause



Select deptno, sum(sal) from emp
group by deptno
having sum(sal) > 17000;

⑤ Step of group by previous
having clause

10000	18000
20000	
30000	
29000	17000
28000	X

PAGE No. _____
DATE _____

it is recommended that only group function should be used in having clause

Step Select deptno, sum(sal) from emp
group by deptno

having sum(sal) > 1700 and sum(sal) < 25000;

of DeptNo sum(sal)
2 → 18000

④ Step of group order by clause {for following query}

Select deptno, sum(sal) from emp group by deptno order by sum(sal);

of DeptNo sum(sal)
2 17000
1 18000

Interview

- 1) select ----- from - ? - explain the select
- 2) where ----- syntax
- 3) group by -----
- 4) having -----
- 5) order by -----

order clause is the last clause in select statement

ex of above

→ Select deptno, sum(sal) from emp

where sal > 7000

group by deptno

having sum(sal) < 10000

order by 2;

\$ select sum(sal) from emp
group by deptno;

O/P

sum (sal)

18000

17000

\$ select sum ~~sum~~ (max (sum(sal))) from emp
group by deptno

O/P

max (sum (sal))

18000

~~matrix exp~~

~~matrix~~
~~exp~~
\$ select deptno, count(*), min(sal), max(sal),
sum(sal) from emp
group by deptno order by ;

O/P

Dept No	count(*)	min(sal)	max(sal)	sum(sal)
10	3	1300	5000	8750
20	8	800	3000	10875
30	6	950	2850	9400

SQL JOINS

DATE 27/8/16

To combine the columns of two or more tables

DEPT

Dept No	DName	Loc
1	TRN	Bang
2	EXP	Delhi
3	MKTG	Cal

wastage of HDD

{ we can but don't }

EMP

CmpNo	ename	sal	DeptNo	Job	Area	DName	Loc
1	Arun	8000	1	M	4	TRN	Bang
2	Ali	7000	1	C	1	TRN	Bang
3	Kiran	3000	1	C	1	TRN	Bang
4	Jack	9000	2	M	1	EXP	Delhi
	Thomas	9000	2	C	1	EXP	Delhi

we can put for

one table

but it is wastage
of HDD

known as redundancy

We can change order of any

* Select dname, ename, from emp, dpt
where dept.deptno = emp.deptno;

<tablename, columnname>

Q.P.

DName ENAME
dept no driving table
emp driven table

DNAMe	ENAME	f will visit after TRN ①
TRN	ARUN	TRN ② TRN ③ the
TRN	Ali	will search for EXP ①
TRN	Kiran	EXP ② they & MKTG
EXP	Jack	if not find
EXP	Thomas	

PAGE NO. _____
driven table → last table → will be fast or slow



→ In order for the join to work faster
Preferably the driving table should
Should be table with lesser no of rows

* select dname, loc, ename, job, say from emp, dep
where dept. deptno = emp. deptno;

* select * from emp, dept
where dept. deptno = emp. deptno;
O/P → will show all tables.

* select deptno, ~~deptno~~, dname, loc, ename, job, say,
from emp, dept where dept. deptno = emp. deptno
will give error (same column)

* Select dept. deptno, dname, loc, ename, job, say
from emp, dept where dept. deptno = emp. deptno;

* select dept. deptno, dept. dname, dept. loc, emp. ename
emp. job, emp. say from dept, emp
where dept. deptno = emp. deptno;

more readable / easy to use

~~(< > <= >=, !=)~~

PAGE No. 27 DATE 8/15

we can use
Select dname, sum(sal) from emp, dept
where dept.deptno = emp.deptno
group by dname;

{ having ----- }
{ ordered by ----- } we can use next to
above code

~~Op~~ DName sum(sal)
TRN 18000
EXP 17000

~~Op~~ Equijoin (Natural Join)

- Join based on equality condition
- Shows matching rows of both table
- most frequently used join
- use → \bowtie joining combine two table.

~~Op~~ Inequijoin (Non-Equijoin)

- join base on inequality (\neq)

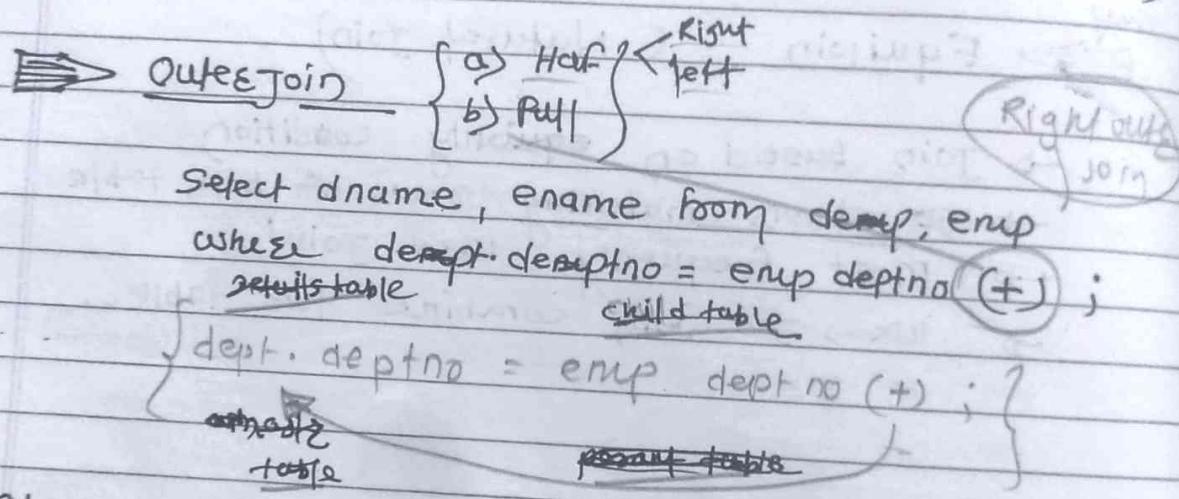
* select dname, dname, ename from emp, dept
where dept.deptno != emp.deptno;

~~Op~~ DName EName
TRN Jack
TRN Thomas
EXP Arun
EXP Ali
EXP Kisan

MKTG	Arun
MKTG	Alli
MKTG	Viraj
MKTG	JACK
MKTG	Thomay

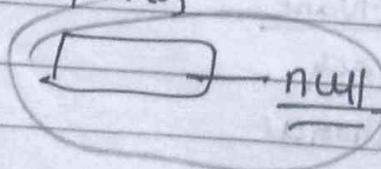
→ Shows non-matching rows of both the
(internally nested for loop) tables

→ practice use → Exception report
(who owe the customer who not payment)



10

DName	EName
TRN	Arey
TRN	AGIL
TRN	Kiru.
EXP	JACK
EXP	Thomas
MKTG	



→ shows matching rows of "both" table
plus

Non matching ^{rows} table of "outer" table

→ outer table ← table which are on
opposite side of (+)

→ use → Master detail Report
(parent child report)

* select dname, ename from dept, emp
where dept.deptno (+) = emp.deptno ;
left outer join

<u>Dname</u>	<u>ename</u>
TRN	Arun
TRN	Ali
TRN	Kiran
EXP	Jack
EXP	Thoum
.	SCOTT

Outer join

a) Half outer join

i) Right outer join
ii) Left outer join

base on do which

b) Full outer join

* (select dname, ename from emp, dept
where dept.deptno = emp.deptno (+))

whole
common

Union

Select dname, ename from emp, dept
where dept.deptno (+) = emp.deptno ;

Op2

<u>DName</u>	<u>EnName</u>
TRN	Ady
TRN	Ali
TRN	Kizay
EXP	Jack
EXP	Thomy
MKTG	
	scott

→ shows matching rows of both the tables
plus non matching rows of both tables

(+) → only works in oracle

~~not for MySQL~~

→ ANSI syntax for full outerjoin : (MySQL)

* Select even dname, ename from full outer
join dept on (dept.deptno = emp.deptno);

Ansi syntax for Right outerjoin :-

Select dname, ename from Right outerjoin
on (dept.deptno = emp.deptno);

Ansi syntax for left outerjoin :- (MySQL)

Select dname, ename from emp left outer
join dept on (dept.deptno = emp.deptno);

MySQL full

To create full outerjoin in MySQL

Select dname, ename from Right outerjoin
on (dept.deptno = emp.deptno)

Union

Select dname, ename = emp.deptno from
left outerjoin on (dept.deptno = emp.deptno);



Innner join

(by default every join is an innner join)
putting ~~plus~~ (+) sign or using the "outerjoin"
keyword is what makes an outerjoin

Cartesian join

(Cross join)
(Join without where clause)

Select dname, ename from emp, dept;

Ex Searching // Extracting the rows

dept - driving table
emp - driven table

* select dname, ename, from emp, dept ;

<u>QP</u>	DName	Ename
TRN	Arun	
TRN	Ali	
TRN	Kiran	
TRN	Jack	
TRN	Thomas	
EXP	Arun	
EXP	Ali	
EXP	Kiran	
EXP	Jack	
EXP	Thomas	
MKTG	Arun	
MKTG	Ali	
MKTG	Kiran	
MKTG	Jack	
MKTG	Thomas	

→ Show all combination

→ driving table have less no of row
for speed up

→ use → printing (e.g marksheet)
payroll

Important

Self join

- Jointable to itself (Slowest join)
- used when parent column and child column both are present in the same table
- based on accuracy (Eg: Situation)

Select a.ename, b.ename from emp b, emp a
where a.mge = b.emp_no ;

A.ename B.ename

Arjun	Jack
Ali	Arun
Kiran	Arun
Thomas	Jack

Senior Yrs

A

Enam.
Arjun
ali
Kiran
Jack
Thomas

Maq

B

EmpNo HR Enam.

1	Arjun
2	Ali
3	Kiran
4	Jack
5	Thomas

Use → Folder into Folders (Ex: Create folder)
Sub folder

Frame	Fname	Mname	Sname
1	C:\	Q	
2	windo	2	
3	MS OFFS	2	
	Nitin	1	

① DEPTHEAD		② DEPT		
DEPTNO	DHEAD	DEPTNO	DNAME	LOC
1	Arun	1	TRN	bby
2	Jack	2	EXP	Plh
		3	MKTG	cst

* Select dname, ename, dhead from emp, dept, depthead
 where depthead.dept no = dept.dept no
 and dept.deptno = emp.deptno;

O/P

DNAME	ENAME	DHEAD
TRN	Arun	Arun
TRN	Ali	Arun
TRN	Kiran	Arun
EXP	Jack	Jack
EXP	Thomas	Jack

→ nested for loop upto 3 level

~~8~~ TYPES OF RELATIONSHIP

- 1 : 1 (Dept : Depthead) or (Depthead : Dept)
- 1 : many (Dept : Emp) and (Depthead : Emp)
- many : 1 (Emp : Depthead) and (Emp : Depthead)
- many : many (Emp : Project) or (Project : Emp)

Ex - 1 - 4
Assignment - 1 - 8 my SQL
→ Oracle

PAGE NO. 1
DATE 27/8/16
Intersection table
Project-Emp

Projects

projNo	projName	custName
P1	PPS	ICICI
P2	AMS	Margu
P3	Macro	Bnp
P4	GAS	Deloitte
:	:	:

Proj No	Grp No
P1	1
P2	2
P2	1
P2	3
P2	4
P3	5
P3	1

* Select projname, ename from projects-emp, projects, emp where projects-emp.empno = emp.empno
and projects-emp = projects.projno;

- ① What is normalization & types.
- ② difference b/w function & stored procedure
- ③ Triggers in SQL & types of triggers
- ④ difference b/w delete & truncate
- ⑤ ACID property
- ⑥ Type of join
- ⑦ Why index & types of Index
- ⑧ what is view
- ⑨ difference between & group by
- ⑩ In SQL → No of emp start with 'A'
→ To get 3rd max salary
⇒ → To display current date.
→ Construct
→ diff b/w primary key & uniq key

Normalization →

→ It is concept of table design.
→ How what table to create, column, datatypes, width, constraints

Advantage →

1) data redundancy

(Avoid unnecessary duplication of data)

2) Reduce the problem of insert update delete

3)