

10/11/24



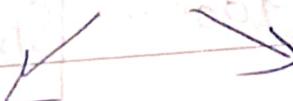
SQl

9 am

- 10 Database :- Collection of data is called database
P.BMS - Software applications to manage our data
- 11 User → Software Application (programming lang) → database

1 pm

Database



- 2 Relational
3 Non-Relational
4 (Use tables to store data)
5 → MySQL
→ Oracle
→ MariaDB
6 → MongoDB
Value
pairs
etc

7 Why SQL :-

- 8 Becoz - We need a language to interact with databases.

Notes → So we use SQL to interact with P.B, do some C.R.U.D. operations on the data.

DQL
SELECT

DML
INSERT
UPDATE
DELETE



DDL
CREATE
ALTER
DROP
TRUNCATE
RENAME

9 am Then ~~why~~ what is MySQL?

- 10 → MySQL is specific Relational database Management system (RDBMS) (Software) which uses SQL as its Querying language.

12 History of SQL: - (Structured Query Language).

1 pm ↳ SQL originated in 1970s from IBM's research on relational databases. It started as SEQUEL (Structured English Query Language)

3 Later renamed SQL due to trademark issues.

4 SQL - It's a programming language used to communicate & manipulate data in databases.

5 It helps users in performing CRUD (Create, read, update, & delete).

6 SQL allows users to perform variety of tasks related to databases.

i) Retrieving data: - Extracting precise info. from a database through queries.

(ii) Manipulating Data: - adding, modifying, removing records within a database.

Notes: (iii) Defining data: - creating & adjusting structure of a database including tables, views & indices.



9 am (iv) Controlling data :- Managing database access
By granting or revoking permissions

Installation of MySQL

MySQL server :- Database server where data is stored, managed & accessed

MySQL workbench :- It is a visual tool which is used for database design, development, administration & management. It provides a user interface (UI) to interact with MySQL server.

Types of SQL Commands

SQL commands are divided into different categories Based on their functionalities :-

1) Data Query language (DQL) commands

2) Data Manipulation Language (DML)

3) Data definition Lang (DDL) - define schema/structure of db

4) DCL data control language commands

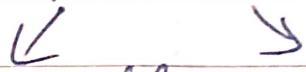
5) Transaction control language commands



- 9 am 1) DQL - Used to retrieve data from database
Command used is Select. (SELECT)
- 10 2) Data Manipulation Language (DML) - Used to manipulate data stored in database.
Commands :- INSERT, UPDATE, DELETE
- 11 1 pm 3) DDL Command - Used to define structure & schema of database :- Commands used are :- CREATE, ALTER, DROP, TRUNCATE, RENAME.
- 2 4) DCL :- Data Control Language :- It deals with control & security of data with the table.
Commands are GRANT, REVOKE
- 5 5) Transaction Control language :- TCI Commands COMMIT, ROLLBACK, SAVE POINT
use to manage transaction within a database.

Creation of Database

8 Database :- School (DB)



Notes

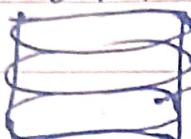
Student Table
~~~~~  
schema

Teacher Table  
~~~~~  
Schema

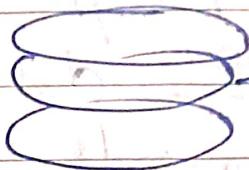
Attributes Entered
10 rows & Col

10 am We create db in a server : First create server.
Multiple dbs in a single server ; can create multiple servers ! tables in a db.

11 DB



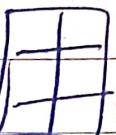
School



Hospital



12 Course table



Patient

2 Creation of db steps:-

1) choose a DBMS (Database Management Systems)
like RDBMS (MySQL).

3) Create a new database

2) Connect to server using command-line tool or GUI.
4) Once the db is created, can use the USE statement
tables in the db.

5) Create Tables & Insert data.

6) We use the CREATE database statement to
create a new db ; end with semi-colon . These
commands are not case sensitive

7) Command :- CREATE DATABASE dbname;

Notes Also to avoid error can use :-

Command :- CREATE DATABASE IF NOT EXISTS dbname;



IS NOT EXISTS, and IF EXISTS clauses are commonly used in conjunction with CREATE TABLE and DROP TABLE statements to avoid errors.

11. Deletion of Database:-

12. We use the DROP Database statement to delete a db. It deletes entire db, including all data, tables & other objects. DROP is a DDL Command

13. `DROP DATABASE dbname;`

14. Also to avoid errors can use:-

`DROP DATABASE IF EXISTS dbname;`

5. Using a database

In server if multiple db are there then by using USE stat. we can use that db. like:-
we can use the USE Statement

7. `USE database;`

8. `COMMAND:- USE DATABASENAME;`

Showing all the database:-

Notes: Use the SHOW DATABASE Statement to see all
me, the database present in a server.
COMMAND:- `SHOW DATABASES;`



9 am

Creating a Table

10

Use Create Table statement to create table in db.

11

Command :-

12

Create Table TableName (column datatype1 constraint1,
Column datatype2 constraint2, ...)

1 pm

Create - DDL Command

Ex:-

CREATE TABLE Employee (EmpID INT primary key, EmpName varchar(20));

4

5

Inserting Values into TABLE

INSERT :- SQL Command

Insert into tableName (Column1, Column2, ...)
VALUES (Value1, Value2, ...);

8

To see all the values in table :-

Select * from tablename
Column

To see all the values Use :- Select * from tablename



9 am * Seeing Tables in a given database

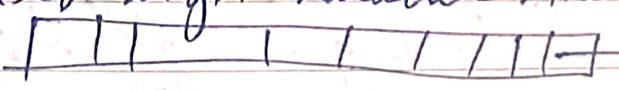
SHOW TABLES

10 It helps us to see all the tables in a given db.

Data types in SQL

11 Data types are used to specify the type of data that a column can store.

Numeric	Character/String	Date & Time	Boolean	Binary
Integer/INT	CHAR (n)	Date	Boolean	Binary
SMALLINT	VARCHAR (n)	Time		
BIGINT	TEXT	Datetime		
DECIMAL		Timestamp		
FLOAT				
DOUBLE				

12 char (n) - Any fixed length character strings can be stored (0 - 255)  Block of Memory

13 It's not good to use char bcoz if we specify char(10) & tell name as AARU then in 10 blocks 6 gets wasted so its not recommended.



- 2) VARCHAR(n) :- Variable-length character strings can be stored (0-255). It is best bcz it does not hold memory & then no memory gets wasted
- 3) TEXT - Variable-length character with no specified limit as above in varchar from 0-255. It has no limit.
- 4) DATETIME/STAMP :- Used for storing date & time values (yyyy-mm-dd hh:mm:ss)
- 1 pm Command :- Create table tablename (Time TIMESTAMP);
- 5) BOOLEAN :- Used to store a true/false value
Command :- CREATE TABLE Tablename (Boolean Boolean);
- 6) BINARY (n) - Used for fixed-length binary data. Memory wasted
- 7) VARBINARY (n) - Used for storing variable-length binary data like in case of Varchar, no memory waste,
- COMMAND :- Varbinary (2) ↗ 0,1,00,01
- 8) BLOB (Binary Large Object) - Used for storing large amounts of binary data (varbin)

Notes Create Table Tablename (data BLOB)

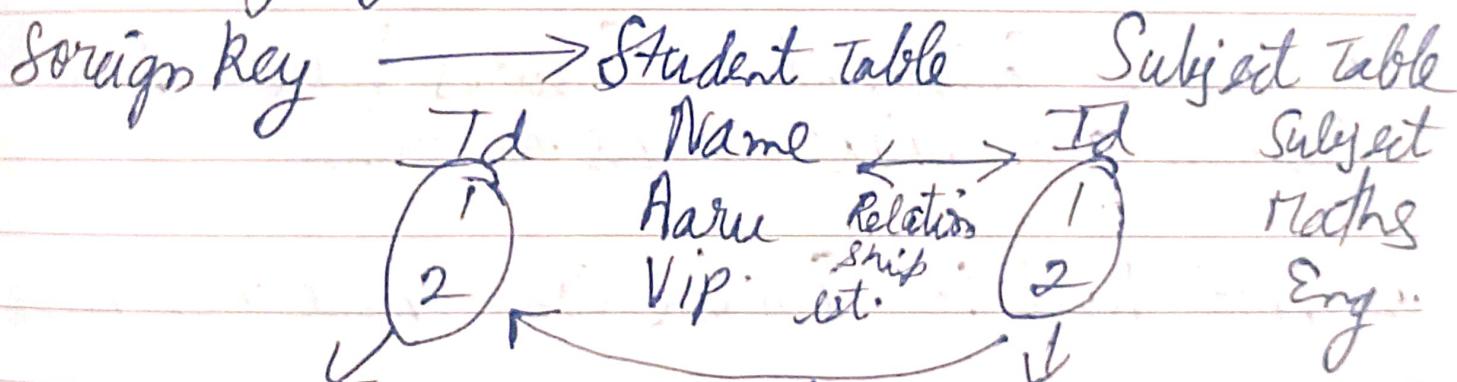


9 am Keys in SQL :-

- 1) Primary key :- is unique identifier for each record in the table. It ensures that each row can be uniquely identified → Unique + Not Null.
- for eg :- key id is unique, phNo. can have not null values
- primary key - id

- 1 pm 2) Foreign key :- is a field in a table that refers to the primary key of another table. It establishes relationship b/w tables.

3) Primary key $\xrightarrow{\text{Always}}$ Unique + Not Null



7) Primary Key ✓ Secondary key
8) taking reference from student table

9 am ♦ Referenced Table - Table having primary key

10 Referencing Table - Table having foreign key

11

12 ♦ Constraints in SQL

1 pm Refines Rules or conditions that must be satisfied by the data in table

2 Common Constraints include Uniqueness, Nullability, Default Values etc.

3 1) Unique Constraint :- Ensures Values in a Column are Unique across the table

4 2) Not Null :- Column Can't have null Value

5 3) Check :- Enforce conditions to be true for each row

6 4) Default :- Provide default Value for columns if no value is specified.

7 Notes 5) Primary Key 6) Foreign key

9 am (NOTE):- Always write tablename in small letter
it doesn't throw error but if write in above
letter then also MySQL consider it as in small letter.

Primary Key

1 pm Create Table School

(stud-id int primary key)

unique, stud-name varchar(20));

stud-id	stud-name	School Table
1	aaru	
2	vip	

Base Reference / Parent Table

Course-id	Course-name	Stud-id	— Course Table
25	Maths	1	
30	Eng.	1	
40	Hindi	2	

Referencing / Child Table

→ Primary key :- Not Null, Unique, Non-Repeated

→ Foreign key - Establish Relationship b/w tables

Foreign key

Create Table Course

(Course-id int not null,

Course-name varchar(20),

Stud-id int, foreign key

(Stud-id) references

School (stud-id));

- 9 am 8 Foreign key : - helps to perform operations related to the parent table, such as joining tables!
- 10 Ensure referential integrity (Any update in primary key must be done in foreign key or in child table also)

12 QUERY :-

```
CREATE TABLE tablename (
    childID INT Primary key ,
    baseID INT , foreign key (baseID)
    REFERENCES basetablename (baseID));
```

3 Cascading is Foreign key .

4 To Maintain referential integrity means to update anything in pkey (Base table) must get updated in child table .

6 → Cascading are set of Rules which tell what actions to be taken automatically when a referenced row in parent table is modified / deleted .

8 1) Cascade - if a row in parent table is updated / deleted , all related rows in child table will be automatic update .



9 am 2) Set Null - If row in parent-table is updated, all foreign key values in child-table will be set to null.

10 3) Restrict/No Action : - Blocks modification of referenced row in parent-table if related rows exist in child-table & Maintains integrity.

12 2 ways to Achieve

1 pm → Cascading in foreign key.

2 1) On Delete Cascade : - It indicates that if row in the parent-table is deleted all corresponding rows in the child-table will be deleted.

4 QUERY :-

5 CREATE Table TableName (child_id INT primary key,
6 baseID INT, foreign key (Base-ID) References
baseTablename (Base-ID)
7 ON DELETE CASCADE);

8 2) On Update Cascade - If row in parent-table is updated then all corresponding rows in child-table will be updated as well.

Notes . Create Table Child-table (child_id INT primary key ,
base_id INT, foreign key (Base-ID) References
base Table Name (Base-ID) ON UPDATE CASCADE);



9 am

SQL : Requirements :-

SQL (Programming lang) perform CRUD

10

C → Create

R → Read

U → Update

D → Delete

11 pm

Update Commands :- Used to modify existing records in table. If a get safe mode error while executing gives run query as
SET SQL_SAFE_UPDATES = 0;

Main Update QUERY IS :-

5 UPDATE Table_name .

SET columnName1 = Value1 (to be set),

6 columnName2 = Value2 (to be set),
where conditions ;

>Delete Command :-

in SQL Used to remove records from a table.

Notes : DML QUERY

Delete from table_name
where condition ;



8. Select Command :- SQL DQL - data Query lang.

9. To Read Data or retrieve data from database.

10. To retrieve data present in specific column in table :-

11. \Rightarrow Select col1, col2 . . . from table-name

12. \Rightarrow Select * from table-name \Rightarrow Used to retrieve all data present in table.

1 pm
Selecting data Using Where Clause :

2. WHERE Clause : - filters rows Based on specific conditions.

3. Query : - Select col1, col2 from TableName where conditions ;

4. Select age from employee where age \geq 20 ;

5. Alter Command : - DDL Used to modify (change) existing database objects such as tables, indices or constraints (Schema)

6. Alter mostly used in addition of new column, deletion, modifications.

7. Schema - Provide all column names or characteristics of table is called schema.



9 am

Alter Command

10) Add column

11) QUERY:-

12) Alter Table TableName
ADD columnName datatype Constraint;

1 pm

13) DROP a column

14) QUERY:-

15) Alter Table tableName
DROP column columnName;

16)

17) 3) Modify the datatype of existing Column

18) Modify clause :- Used within an alter Table in SQL.
allow us to change definition of existing col' in a table.

Notes

Query:- Alter Table tableName
Modify columnName newdatatype;



9 am 6) change Name of Existing column - used to change name or data type of a column within a table.

10 QUERY :-

11 Alter Table TableName

12 CHANGE OldColName newColName newdatatype;

1 pm Eg:- Alter table Employee
change age emp-age INT;

2 OR.

3 5) Rename the name of existing Column - Rename of
4 Col, table, index or constraint.

5 QUERY :- Alter Table Table name

RENAME ~~old~~ column OldColName TO NewColName;

6 To Rename Table name Query:- Rename Table OldTableName
7 To NewTableName;

8 To Rename (Database Renaming) :-

Query:- Rename database OldDatabaseName to NewDatabaseName

Notes



④ Truncate Command:

9 am

This removes all rows from given table, leaving table empty but preserving its structure.

Query:- `Truncate Table TableName;`

⑤ Difference b/w Truncate, Delete & Drop Command.

1 pm

Truncate

Delete

Drop

- 1) Remove all rows from table
- 2) `TRUNCATE Table Table Name;`

Removed specific rows from table based on condition
Delete from tablename where conditions;

Used to complete remove table
DROP, tableName
TABLE

⑥ Using Distinct (Unique) to retrieve Unique Values

Distinct keyword used within select statement to retrieve unique values from column or combination of columns.

Query:- `Select distinct col1 from TableName` → retrieves list of unique values for col1

Notes `Select distinct col1, col2 from TableName` → returns unique combinations of col1 & col2.



9. Operators in SQL :- Perform Operations on data

10. Query : Select All Col 2-5 from table Name where cond;
(Logical Operator)

11. Types of Operators :-

12. Aggregate functions perform some operation on a set of rows & then return single value summarizing data. Used with SELECT Statement to perform calculations.
→ These are collected or collected.

13. Types of aggregate functions are :-

- 1) COUNT() - It count no. of rows in table / no. of non-null values. Count how many things are in list.
- 2) SUM()
- 3) AVG()
- 4) MIN()
- 5) MAX()
- 6) GROUP_CONCAT()

Aggregate Functions

9 am

1) COUNT()

10 Query :- Select COUNT (name) from Employee;
this will tell no. of employees in Company.

11 2) SUM() - calculates the sum of all values in a

12 Numeric Columns. This adds up all the

number in a list. (non numeric value like name)

NOTE:- If going to add non numeric value like name then it will give result as error

Query :- Select SUM (Salary) from Employee

2) (This will tell total amount company is
paying to its employees).

3) AVG() - It computes the average of all values

in a Numeric Column. finds average or
"middle" number of all the numbers in a
list,

6) Query :- Select AVG (Salary) from Employee :>

this will tell the avg amount company is
paying to its employees.

8) If going to avg non-numeric values like
name or string then it will not give any
error, it will execute & the query
give result as 0.

Notes



4) MIN() - It helps to find smallest number in a list.

Query = select min(salary) from employee

5) MAX() \Rightarrow Highest number in a list

Query = select max(salary) from employee

NOTE

Min() & Max() - Not only perform operations on numbers, we can use min and max as aggregate as well like ~~min~~ min(name) \Rightarrow aarun
Max(name) \rightarrow vip

Grouping Data with the GROUP BY CLAUSE

GROUP BY CLAUSE :- Used to group rows that have same values into together. help to organise data into groups that we can do calculations, like finding totals or averages for each group.

9 am This query retrieves first n rows from the

10 Query :-

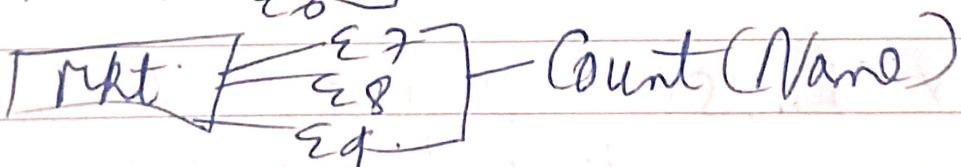
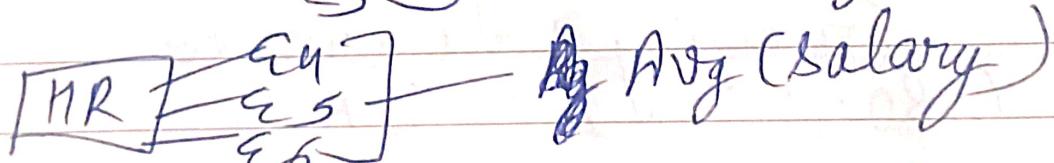
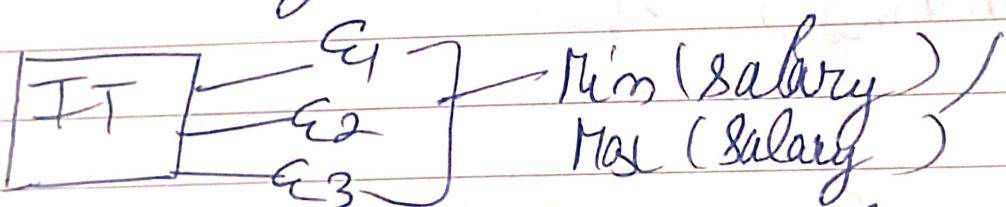
11 Select col1, aggregates (Col 2)
12 from tablename group by col1;

13 e.g. :- Select dept, Avg(Salary) AS avgsal
from Employee group by dept.

14 # AS acts as alias or the name of the result column.

15 Consider group by clause with select :- In Industry

16 There are so many dept such as IT, HR etc



20 So in above all three dept we want to find

Notes separately that is each dept what is the
max & min salary in each dept of employee
so to group dept & then find aggregate func we use
GROUP BY CHOICE.

9 am GROUP BY CLAUSE Eg:-

10	dept	Salary
11	IT	1100
12	IT	1000
1 pm	HR	500
	IT	2500
	Mkt	1000
	HR	200

2 Select dept, avg(salary) as result from
 3 employee group by dept.

Result:

4 [IT :- 1533.3, HR - 350, Mkt = 1000]

5 HAVING Clause
 6 → works on aggregated data

7 HAVING clause - is just like the "where" clause
 8 but the main difference is it works on
 aggregated data used with the group BY
 clause. helps to filter groups based on given
 conditions.



10 am Query :-

10 Select col1, col2 aggregate fun (col3)

from tablename

11 Group BY col1 col2
HAVING Conditions ;

12 Eg :- Select dept, avg(salary) as result

from employee

GROUP BY dept

2 HAVING result > 1500 ;

3 dept \rightarrow IT - Avg Sal > 1500 ✓ show result

\rightarrow HR - Avg Sal > 1500 ✓

4 \downarrow Mkt - Avg Sal < 1500 - not show result output .

5 Phle to dept vali col me se three dept ko

6 group kiya jayga fir unke avg salary nikalei jayegi har ek group ki & woh Bad

7 jis bhi dept ki avg sal 1500 se zada hogi wahi output me show hoga otherwise nahi .



Difference Between Where and Having clause

Where

Used to filter rows from result based on cond. apply to row before aggregation.

Used with Select, Update or delete Commands

Select * from tablename where cond.

Having

Used to filter rows from result based on cond. apply after aggregation.

Used with group by & aggregate funcns

Select col1, col2 agg func from tablename group by col1 col2 having cond.

General Order of SQL Commands like before applying this u have to apply its above command

Sno	Command	Usecase
1	SELECT	Retrievce from database
2	FROM	Identify table
3	WHERE	filter rows based on cond.
4	GROUP BY	group rows that have same values
5	HAVING	filter grp based on some cond.
6	Order by	sort the result set either asc/desc
7	LIMIT	limit no. of rows returned



9 am

Joins in SQL

10

- 11 Joins used to combine rows from 2 or more tables based on related or shared or common columns b/w them.
- 12 4 types of joins including INNER Join, left join, right join, full join, self join, cross join.

1 pm

for eg : - 2 tables are there like :-

Student Table		
Age	Std-name	Std-id
14	Aaru	4
12	Vip	5

Course-id	Course-name	Std.
1	Maths	5
2	Eng	4

Q) Is foreign key important for performing joins?

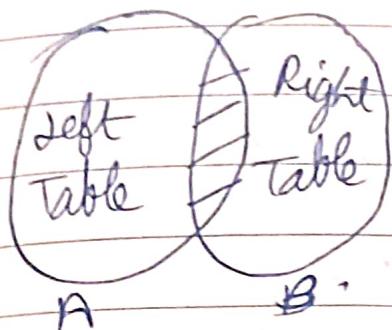
Ans) joins can be performed based on any columns that establish a relationship b/w tables, not just FK constraints, so it's not necessary, only need shared column.

Notes



9 AM Types Of Joins in SQL :-

1) Inner join

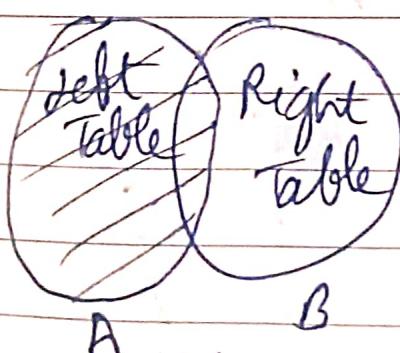


A		B	
Rollno	Name	Role	Course
1	Ram	3	Hindi
2	Rahul	2	Eng
3	Aaru	4	Maths

1 PM We print output as the common rows which present in both the tables A & B.

Output :-	Rollno	Name	Course-name
2	Rahul	Eng	
3	Aaru	Hindi	

2) Left Join / Left Outer Join



left

Notes :- In left join me left me jo the same values h
vo output me likhi asti h aur jo common values h
left & right table me, right table se vo values
nxti h.

9 am

	A		B	
10	Roll_no	Name	Roll_no	Course-name
3		Aaru	1	Eng
4		Vip	4	Hindi
5		Scooby	5	Doggy-long

12

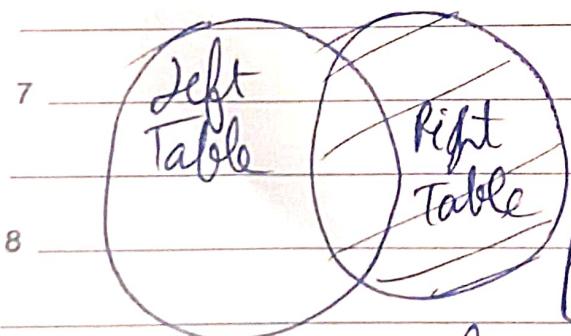
Output:- after applying left join

	Roll_no	Name	Course-name
1	3	Aaru	Null
2	4	Vip	Hindi
3	5	Scooby	Doggy-long

4. 3 roll_no not in the Right Table (course me)
 that's why we're value null aai.

5

• Right Joins / Right Outer joins



Roll_no	Name	Roll_no	Course
1	Aaru	3	Eng
2	Scooby	2	Doggy-long
3	Vip	4	Bengali

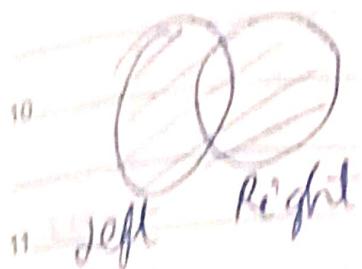
Notes

Output:-

Roll_no	Course	Name
3	Eng	Vip
2	Doggy-long	Scooby
4	Bengali	Male



4) full join / full outer join



Roll_no	name	Roll_no	Course
1	Ram	2	Hindi
2	Rahul	3	Eng
3	Aaru	4	Maths

Output	Roll_no	Name	Course
	1	Ram	Null
	2	Rahul	Hindi
	3	Aaru	Eng
	4	Null	Maths

It shows all the value present in left and right table with the common values in both the table as well.

5) Self Join

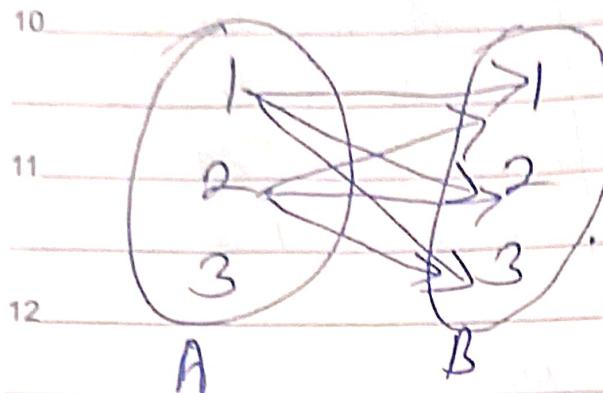


Performing join on itself

Notes

9 am

6) Cross Join



In Cross join table the one row will do the Cartesian product with another table like 1 will go & match output with 1, 2 & 3 same as 2 with

	A	B		
1 pm	Roll-no	Name	Roll-no	Course
2	1	Ram	2	Hindi
3	2	Rahul	3	Eng
	3	Kare	4	Math

In Cross join the output will be 3×3 having 9 rows like 3 in A & 3 in B rows so output will be $3 \times 3 = 9$ rows.

Notes

7 Select customer.customer-id, customer.last-name, fruit.cart-fruits
from customer inner join fruit on customer.customer-id =
fruit.customer-id;



9 am 8 Inner Joins :- help us in getting the rows that have
10 matching values in Both tables, according to
11 gives join conditions.

11 Query :- Select columns
12 from table1name
13 inner join table2name
1 pm ON table1.colname = table2.colname;
2 *table1.colname which is similar in both
the tables*

3 → It only returns rows where there is matching id in
both tables.

4 8 Left Outer join / Left join - Used to fetch all records
5 from left table along with matched records from
6 right table.
7 if there are no matching records in right table, NULL
values are returned for the columns of right table.

8 Query :- Select Columns
from table1name is the left table
left join table2name is right table
ON table1.colname = table2.colname;

Notes

9 am Left join sample :-

Customer Table

	Customer-id	Customer-name
10	1	arun
11	2	scoby
12	3	pol
	4	fur

Fruit Table

Customer-id	Customer-name
1	Mango
5	litchi
4	Banana
2	plum

1 pm
Query :- Select * from customer left joins fruit on
customer.customer-id = fruit.customer-id.

Output :-

Customer-id	Customer-name	Customer-id	Cust-fruits
1	arun	1	Mango
2	scoby	2	plum
3	pol	Null	Null
4	fur	4	Banana

6
7 Right joins / Right Outer joins :- Used to fetch all the records from right table along with matched records from left table. If there is no matching records in left table, NULL values are returned for the columns of left table.



Query :- Select columns from Table1name \rightarrow left Table
RIGHT JOIN Table2name \rightarrow Right Table
ON Table1.colName = Table2.colName ;

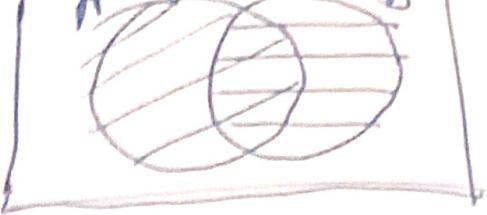
fruit is right Table & customer table is left table

Select * from customer right join fruit
On customer.Customer-id = fruit.Customer-id

Customer-id	cust_name	customer-id	cust_fruit
1	Aarue	1	Mango
Null	NULL	5	Litchi
4	BWC	4	Banana
2	Soby.	2	Plum

FJ = LJ + RJ

- full joins / full outer joins :- It returns the matching rows of both left & right table & also includes all rows from both tables even if they don't have matching rows.
- If there is no match, NULL values are returned for the columns of missing table.



$$\begin{aligned} FJ &= L J + R J \\ &= A + G + B + C \end{aligned}$$

10. In output will get only one time C not twice
11. bcoz we are using Union, give result eliminating
duplicates C twice.

12. * In MySQL, the syntax of full joins is different
1 pm Compared to other SQL databases like PostgreSQL
or SQL Server

→ MySQL doesn't support FULL JOIN keyword
directly. So we use combinations of
left join, right join & UNION to achieve
result.

→ Full JOIN Query:- $A \cup B$ 

5. Select columns
from tablename
left join tablename
ON table1.colname = table2.colname;
UNION

6. Select columns
from tablename
right join tablename
ON table1.colname = table2.colname;

9 am \Rightarrow By considering the sample of customer & fruits on full join the output is :-

10 Output:- Full join / UNION Results

Customer-id	Cust-name	Customer-id	Cust-fruits
1	aaru	1	Mango
2	scoby	2	apple
3	pol	NULL	NULL
4	fir	4	Banana
NULL	NULL	5	litchi

3 \star CROSS JOIN :-

4 It Combines each row of first table with every row of second table .

5 Query :- Select * from table1
CROSS JOIN table2 ;

7 In our sample of customer & fruit table the output rows will be $m \times n = 4 \times 4 = 16$ rows .

8 Some output rows are :-

Customer-id	Cust-name	Customer-id	Cust-fruits
4	fir	1	Mango
3	pol	1	Mango
2	scoby	1	Mango
1			"



9 am
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
988
989
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1919
1920
1921
1922<br

9 am ~~Self join sample table customer-table~~

Customer-id	Customer-name	Mentor-id
1	Aaru	2
2	Scoby	1
3	Pol	1
4	Fur	3

1 pm Query of Self join :-

- ~~2 Select C1.mentor-name, C2.Customer-name from Customer as C1 join Customer as C2 where C1.Customer-id = C2.mentor-id;~~
- ~~4 Select C1.Customer-name as mentor-name, C2.Customer-name as Customer-name from Customer as C1 join Customer as C2 where C1.Customer-id = C2.mentor-id;~~
- 5 Output:

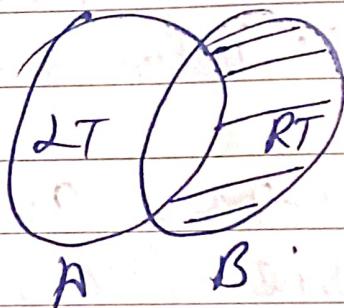
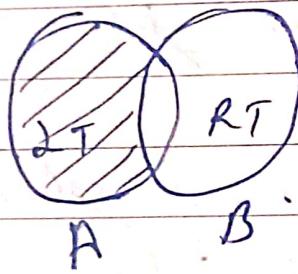
Mentor-name	Customer-name
Aaru	Scoby
Aaru	Pol
Pol	Fur

Exclusive Joins in SQL

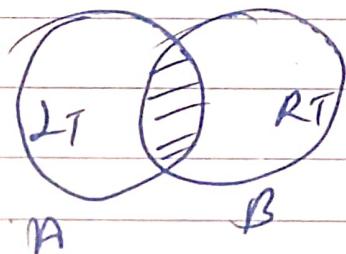
Exclusive joins used when we want to retrieve data from 2 tables excluding matched rows. They are part of outer joins or full outer joins.

Types :-

- 1) Left Exclusive Join
- 2) Right Exclusive Join



- 3) full Exclusive join (A ∩ B)





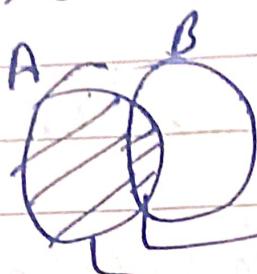
9 am
8 Left Exclusive Joins - When we retrieve records from the left table excluding the ones matching in both left & right table.

10
11 Query :- Select columns from table 1
left joins table 2

12
13 ON table1.colname = table2.colname ;
where table2.colname is NULL;

1 pm
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
33100
33101
33102
33103
33104
33105
33106
33107
33108
33109
33110
33111
33112
33113
33114
33115
33116
33117
33118
33119
331100
331101
331102
331103
331104
331105
331106
331107
331108
331109
331110
331111
331112
331113
331114
331115
331116
331117
331118
331119
3311100
3311101
3311102
3311103
3311104
3311105
3311106
3311107
3311108
3311109
3311110
3311111
3311112
3311113
3311114
3311115
3311116
3311117
3311118
3311119
33111100
33111101
33111102
33111103
33111104
33111105
33111106
33111107
33111108
33111109
33111110
33111111
33111112
33111113
33111114
33111115
33111116
33111117
33111118
33111119
331111100
331111101
331111102
331111103
331111104
331111105
331111106
331111107
331111108
331111109
331111110
331111111
331111112
331111113
331111114
331111115
331111116
331111117
331111118
331111119
3311111100
3311111101
3311111102
3311111103
3311111104
3311111105
3311111106
3311111107
3311111108
3311111109
3311111110
3311111111
3311111112
3311111113
3311111114
3311111115
3311111116
3311111117
3311111118
3311111119
33111111100
33111111101
33111111102
33111111103
33111111104
33111111105
33111111106
33111111107
33111111108
33111111109
33111111110
33111111111
33111111112
33111111113
33111111114
33111111115
33111111116
33111111117
33111111118
33111111119
331111111100
331111111101
331111111102
331111111103
331111111104
331111111105
331111111106
331111111107
331111111108
331111111109
331111111110
331111111111
331111111112
331111111113
331111111114
331111111115
331111111116
331111111117
331111111118
331111111119
3311111111100
3311111111101
3311111111102
3311111111103
3311111111104
3311111111105
3311111111106
3311111111107
3311111111108
3311111111109
3311111111110
3311111111111
3311111111112
3311111111113
3311111111114
3311111111115
3311111111116
3311111111117
3311111111118
3311111111119
33111111111100
33111111111101
33111111111102
33111111111103
33111111111104
33111111111105
33111111111106
33111111111107
33111111111108
33111111111109
33111111111110
33111111111111
33111111111112
33111111111113
33111111111114
33111111111115
33111111111116
33111111111117
33111111111118
33111111111119
331111111111100
331111111111101
331111111111102
331111111111103
331111111111104
331111111111105
331111111111106
331111111111107
331111111111108
331111111111109
331111111111110
331111111111111
331111111111112
331111111111113
331111111111114
331111111111115
331111111111116
331111111111117
331111111111118
331111111111119
3311111111111100
3311111111111101
3311111111111102
3311111111111103
3311111111111104
3311111111111105
3311111111111106
3311111111111107
3311111111111108
3311111111111109
3311111111111110
3311111111111111
3311111111111112
3311111111111113
3311111111111114
3311111111111115
3311111111111116
3311111111111117
3311111111111118
3311111111111119
33111111111111100
33111111111111101
33111111111111102
33111111111111103
33111111111111104
33111111111111105
33111111111111106
33111111111111107
33111111111111108
33111111111111109
33111111111111110
33111111111111111
33111111111111112
33111111111111113
33111111111111114
33111111111111115
33111111111111116
33111111111111117
33111111111111118
33111111111111119
331111111111111100
331111111111111101
331111111111111102
331111111111111103
331111111111111104
331111111111111105
331111111111111106
331111111111111107
331111111111111108
331111111111111109
331111111111111110
331111111111111111
331111111111111112
331111111111111113
331111111111111114
331111111111111115
331111111111111116
331111111111111117
331111111111111118
331111111111111119
3311111111111111100
3311111111111111101
3311111111111111102
3311111111111111103
3311111111111111104
3311111111111111105
3311111111111111106
3311111111111111107
3311111111111111108
3311111111111111109
3311111111111111110
3311111111111111111
3311111111111111112
3311111111111111113
3311111111111111114
3311111111111111115
3311111111111111116
3311111111111111117
3311111111111111118
3311111111111111119
33111111111111111100
33111111111111111101
33111111111111111102
33111111111111111103
33111111111111111104
33111111111111111105
33111111111111111106
33111111111111111107
33111111111111111108
33111111111111111109
33111111111111111110
33111111111111111111
33111111111111111112
33111111111111111113
33111111111111111114
33111111111111111115
33111111111111111116
33111111111111111117
33111111111111111118
33111111111111111119
331111111111111111100
331111111111111111101
331111111111111111102
331111111111111111103
331111111111111111104
331111111111111111105
331111111111111111106
331111111111111111107
331111111111111111108
331111111111111111109
331111111111111111110
331111111111111111111
331111111111111111112
331111111111111111113
331111111111111111114
331111111111111111115
331111111111111111116
331111111111111111117
331111111111111111118
331111111111111111119
3311111111111111111100
3311111111111111111101
3311111111111111111102
3311111111111111111103
3311111111111111111104
3311111111111111111105
3311111111111111111106
3311111111111111111107
3311111111111111111108
3311111111111111111109
3311111111111111111110
3311111111111111111111
3311111111111111111112
3311111111111111111113
3311111111111111111114
3311111111111111111115
3311111111111111111116
3311111111111111111117
3311111111111111111118
3311111111111111111119
33111111111111111111100
33111111111111111111101
33111111111111111111102
33111111111111111111103
33111111111111111111104
33111111111111111111105
33111111111111111111106
33111111111111111111107
33111111111111111111108
33111111111111111111109
33111111111111111111110
33111111111111111111111
33111111111111111111112
33111111111111111111113
33111111111111111111114
33111111111111111111115
33111111111111111111116
33111111111111111111117
33111111111111111111118
33111111111111111111119
331111111111111111111100
331111111111111111111101
331111111111111111111102
331111111111111111111103
331111111111111111111104
331111111111111111111105
331111111111111111111106
331111111111111111111107
331111111111111111111108
331111111111111111111109
331111111111111111111110
331111111111111111111111
331111111111111111111112
331111111111111111111113
331111111111111111111114
331111111111111111111115
331111111111111111111116
331111111111111111111117
331111111111111111111118
331111111111111111111119
3311111111111111111111100
3311111111111111111111101
3311111111111111111111102
3311111111111111111111103
3311111111111111111111104
3311111111111111111111105
3311111111111111111111106
3311111111111111111111107
3311111111111111111111108
3311111111111111111111109
3311111111111111111111110
3311111111111111111111111
3311111111111111111111112
3311111111111111111111113
3311111111111111111111114
3311111111111111111111115
3311111111111111111111116
3311111111111111111111117
3311111111111111111111118
3311111111111111111111119
33111111111111111111111100
33111111111111111111111101
33111111111111111111111102
33111111111111111111111103
33111111111111111111111104
33111111111111111111111105
33111111111111111111111106
33111111111111111111111107
33111111111111111111111108
33111111111111111111111109
33111111111111111111111110
33111111111111111111111111
33111111111111111111111112
33111111111111111111111113
33111111111111111111111114
33111111111111111111111115
33111111111111111111111116
33111111111111111111111117
33111111111111111111111118
33111111111111111111111119
331111111111111111111111100
331111111111111111111111101
331111111111111111111111102
331111111111111111111111103
331111111111111111111111104
331111111111111111111111105
331111111111111111111111106
331111111111111111111111107
331111111111111111111111108
331111111111111111111111109
331111111111111111111111110
331111111111111111111111111
331111111111111111111111112
331111111111111111111111113
331111111111111111111111114
331111111111111111111111115
331111111111111111111111116
331111111111111111111111117
331111111111111111111111118
331111111111111111111111119
3311111111111111111111111100
3311111111111111111111111101
3311111111111111111111111102
3311111111111111111111111103
3311111111111111111111111104
3311111111111111111111111105
3311111111111111111111111106
3311111111111111111111111107
3311111111111111111111111108
3311111111111111111111111109
3311111111111111111111111110
3311111111111111111111111111
3311111111111111111111111112
3311111111111111111111111113
3311111111111111111111111114
3311111111111111111111111115
3311111111111111111111111116
3311111111111111111111111117
3311111111111111111111111118
3311111111111111111111111119
33111111111111111111111111100
33111111111111111111111111101
33111111111111111111111111102
33111111111111111111111111103
33111111111111111111111111104
33111111111111111111111111105
33111111111111111111111111106
33111111111111111111111111107
33111111111111111111111111108
33111111111111111111111111109
33111111111111111111111111110
33111111111111111111111111111
33111111111111111111111111112
331111111111

9 am
On previous page using left joins we not only select left table values but along with intersect value as well.



10 we have to exclude this part &
11 Only show ← this part as
12 Output using

1 pm left exclusive join

2 Query :- of left exclusive join :-

3 select c1.customer-id, c1.cust-name, c2.cust-fruits
from customer as C1 left join fruit as C2
4 ON C1.customer-id = C2.customer-id where
C2.customer-id is NULL;

5 Output :-

Customer-id	cust-name	cust-fruits
3	pol	NULL

7 # Right Exclusive Joins - When we retrieve
8 records from right table excluding the ones
matching in both left and right table.

Notes

9 am Query :- Select columns from table 1

10 Right join table 2

11 ON table1.colname = table2.colname

12 where table1.colname is NULL;

13 Output:-

	Customer-id	Customer-name	Customer-id	Customer-name	Customer-id	Customer-name
1 pm	NULL	NULL	5	litchi		

14 Query :- select * from customers right join fruit

15 on customer.customer-id = fruit.

16 customer-id where customer.customer-id is

17 NULL;

18 \$ Full Exclusive JOIN :- When we retrieve

19 records from right table & left table

20 excluding the ones matching in both left &

21 right table.

22 Query :- Select columns from table 1 left join table 2

23 ON table1.colname = table2.colname

24 where table2.colname is NULL

25 UNION

26 Notes :- Select columns from table 1 right join table 2

27 ON table1.colname = table2.colname

28 where table1.colname is NULL;



9 am
Query :-

10 Select * from customers left join fruit on
11 customer.customer-id = fruit.customer-id where
fruit.customer-id is NULL
12 UNION
13 Select * from customers right join fruit on
14 customer.customer-id = fruit.customer-id
where customer.customer-id is NULL;

Output :-

customer-id	cust name	customerid	custfruit
3	pol	NULL	NULL
NULL	NULL	5	lithi

5 UNION OPERATOR :- Used to combine the
6 results of 2 or more select queries into a
single result set & gives unique rows by
removing duplicate rows.

8 Things to NOTE :- 1) Each select command
with the UNION must retrieve the
same number of columns.

- 9 AM 2) The data types of columns in corresponding positions across SELECT statements should match.
- 10 3) Columns should be listed in same order across all SELECT statements.

11 Query :- Select columns from table 1

UNION

Select columns from table 2 ;

1 pm	id	+	id	=	id
2	1		4		1
3	2		2		2
	3		7		3

4 UNION ALL operator in SQL

5 Used to combine the results of 2 or more select queries into a single result set & gives all rows by not removing duplicate rows.

7 Query :- Select column from table 1

UNION ALL

Select column from table 2 ;

8 Notes	id	+	id	=	id
	1		3		1
	2		3		2
	3		4		3

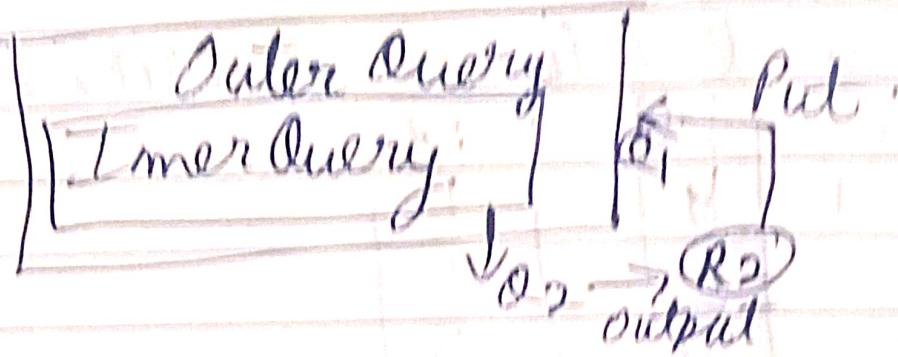


9 am

Q. SQL Subqueries / Nested Queries

- 10 Subqueries / Inner Queries / Nested - It is a query nested within another SQL statement.
- 11 whenever we want to receive data based on whatever we want to receive data based on result of another query we use nested queries.

1 pm



- 3 → Subqueries can be used:-
 - 4 1) With clauses such as SELECT, INSERT, Update or delete to perform complex data retrieval.

6 Query :- Selected Columns, (Subquery)
from tablename;

7 2) WHERE clause to filter data based on condition.

8 Query :- Selected Columns from tablename
WHERE Column name Operator Subquery

Notes



3) FROM clause

Query :- Select * from (Subquery) As allname;

Q) Find all employees who have salary greater than the min salary.

A) To find answer of this ques 1) find min(salary)

2) Select min(salary) from employees.

3) Find salary of employees > Min(salary)

4) Select name, salary from employee where
salary > (Subquery);

Finally, query :- Select name, salary from employee
where salary > (Select min(salary)
from employee);

Q) Find employees who having age greater than min age.

A) i) Min age find :- Select min(age) from employee;

Note:- ii) Greater than :- Select name, age from employee
where age > (Select min (age) from employee);

Q) Print the employees with average age & age of

A) i) employees : - select avg (age) from employees;

ii) select name, age, (select avg (age)) from employees
from employees ;

Q) find the n^{th} highest salary in given dataset :-

Normal function:- 2nd Highest salary

3) select max (salary) from employee
where salary \geq (select max (salary) from
employee);

5) we have write it down bcoz not equal to 1st highest
salary.

6) But if they want to find n^{th} highest then query is

7) \rightarrow limit clause : - used to restrict no. of rows
returned by a query

8) 1) limit n → helps to retrieve max of n
rows from begin of result set

Notes
2) limit m, n → Retrieve specific range of rows
 $m \rightarrow$ no. of rows skip from begin
 $n \rightarrow$ to fetch after skipping

$m \Rightarrow$ no of rows to fetch after skip
like $m=2, n=1$



2 X
3
4
5

rows to skip from begin.

9 am find n^{th} highest salary
Query: \Rightarrow select distinct salary
from table name
order by salary desc
limit $m-1, 1;$
 \uparrow $m \Rightarrow$ shows
 \downarrow after being
skip skipped
 $m-1 = m$, $m = n-1$.

3 * stored procedures - programs that can perform specific tasks based on the stored query.
4 basically a collection of pre-written SQL statements grouped together under specific name.

5 Query :- (to create a procedure)

6 Create Procedure procedureName ()

7 BEGIN

8 Query

END ;

Procedure is like function

↓
Parameters
Nos. para,
eters

Notes Query :- (to call procedure)
CALL procedureName();



• ~~PL/SQL~~ - Stored procedure without parameters :-

• Query :-
Create procedure getallorders ()

• BEGIN
Select * from orders;
END;

• Call getallorders();

• → Stored procedure with parameters :-

• Get Retain the details of the order by id :-

• Query :- Create procedure getallorders (IN id int)

• BEGIN
Select * from orders where id = id ;
END;

• Call getallorders (2);

• VIEWS in SQL :-

• A view is a Virtual table in SQL. It helps in
Notes providing a filtered view of data for
Security purposes.



for a company there is db of employees in which
table

id	Name	password	age
1	John	123456	25

It provide access of this table to anyone as it contains password for security reason we create view

Query :- Create View viewname as
Select Columns from baseTableName (Specify the columns to be included in view)

→ It helps in data abstraction, security & simplify complex queries

→ To see all the data in view

Query :- Select * from Viewname;

→ To drop a view

Query :- Drop View if exists Viewname;

* Case And If In SQL :-

Notes → Case - Allows us to perform conditional logic within a query. It can be used in both



9 am

select & Update Statement to evaluate Conds & return
10 Specific Values based on those Conds

11 Query :- Select columns,
Case

12 WHEN Cond1 THEN result1 WHEN
Cond2 THEN result2 . . . ELSE resultN
1 pm END

2 Ex :- Select city, user_name, Case when

Query City = "mumbai" Then "costly city"

3 When city = "pune" Then "chalega" else
"cheap city" END as city_name/result

4 from users ;

Output :-	City	username	city_name
6	Mumbai	Aaru	Costly city
7	Pune	Vip	chalega
8	Delhi	Scoby	cheap, city
9	Jarawla	Jop	

Q) Categorise the Students on Basis of their percentage
to TOP, Pass & fail in new Column Category.

Notes



Query :- Select Stud-name, Stud-id, marks,
Case
AND
When marks = BETWEEN 80 AND 100 THEN
("Excellent")
When marks BETWEEN 40 AND 79 THEN
("PASS")
When marks // 39 AND 20 THEN ("FAIL")
ELSE ("DOUBLE FAILED")
END AS Grades
from Student;

Q). Students have got some grace marks so update
their grades, where its A update to A+ &
where its B update to A.

A) Query :- Update grades stuck
set grade = A+

Update student

SET grade = Case

WHEN grade = 'B', Then 'A'
WHEN " " = 'A', Then 'A+'

END;

9 am

- ~~IF~~ :- Used to return one of 2 values depending on whether a condition is true or false. It is not supported in MySQL but in MySQL supports

Query:- ~~IF (condition, value-if-true, value-if-false)~~;

1 pm

Select sid, name, percentage,
if (percentage > 90, "TOP", IF (percentage
BETWEEN 89 AND 34, "PASS", "FAIL"))
AS CATEGORY
FROM Student;

- Q) Swap all 'f' with 'm' value & vice versa with single update statement & no intermediate tables.

Query:- Update Employee

Set gender = if (gender = 'm', 'f',

8