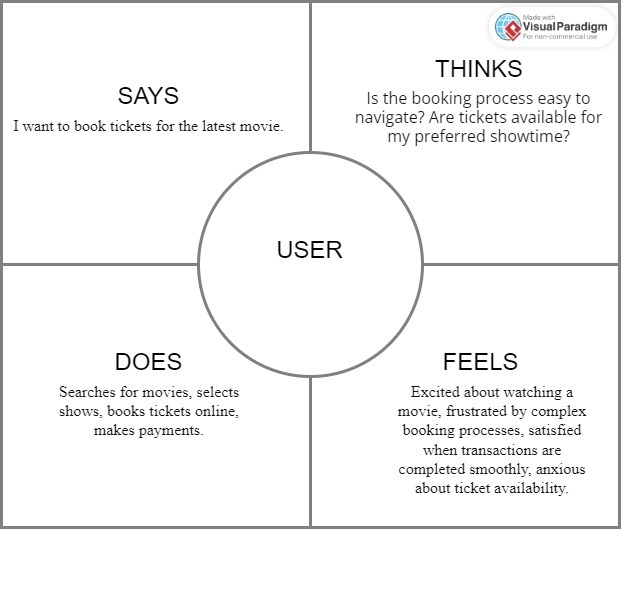


**EMPATHY MAP:-**

****

**JOINING TABLES:-**

Joining tables in SQL allows you to combine data from two or more tables based on a related column between them. This related column is typically a primary key in one table and a foreign key in another. Here's the basic syntax of a SQL join**:**

**SELECT columns**

**FROM table1**

**JOIN table2 ON table1.column = table2.column;**

**There are several types of joins in SQL:**

**INNER JOIN:** Returns rows when there is at least one match in both tables.

**Admin table:**

+------+------------+

| aid | password |

+------+------------+

| 3001 | password1 |

| 3002 | password2 |

| 3003 | password3 |

| 3004 | password4 |

| 3005 | password5 |

| 3006 | password6 |

| 3007 | password7 |

| 3008 | password8 |

| 3009 | password9 |

| 3010 | password10 |

| 3011 | password11 |

| 3012 | password12 |

| 3013 | password13 |

| 3014 | password14 |

| 3015 | password15 |

**TICKET TABLE:**

**+---------+------+------+------+----------+------------+------------+-------------+**

**| show\_id | cid | tid | aid | offer\_id | hallnumber | show\_date | seat\_number |**

**+---------+------+------+------+----------+------------+------------+-------------+**

**| 200 | 1011 | 1 | 3001 | DISC001 | 1111 | 2024-03-17 | 101 |**

**| 201 | 1012 | 2 | 3002 | DISC002 | 1112 | 2024-03-18 | 102 |**

**| 202 | 1013 | 3 | 3003 | DISC003 | 1113 | 2024-03-19 | 103 |**

**| 203 | 1014 | 4 | 3004 | DISC004 | 1114 | 2024-03-20 | 104 |**

**| 204 | 1015 | 5 | 3005 | DISC005 | 1115 | 2024-03-21 | 105 |**

**| 205 | 1016 | 6 | 3001 | DISC001 | 1116 | 2024-03-17 | 106 |**

**| 206 | 1017 | 7 | 3002 | DISC002 | 1117 | 2024-03-18 | 107 |**

**| 207 | 1018 | 8 | 3003 | DISC003 | 1118 | 2024-03-19 | 108 |**

**| 208 | 1019 | 9 | 3004 | DISC004 | 1119 | 2024-03-20 | 109 |**

**| 209 | 1020 | 10 | 3005 | DISC005 | 1120 | 2024-03-21 | 110 |**

**| 210 | 1023 | 11 | 3001 | DISC001 | 1121 | 2024-03-17 | 111 |**

**| 211 | 1024 | 12 | 3002 | DISC002 | 1122 | 2024-03-18 | 112 |**

**| 212 | 1025 | 13 | 3003 | DISC003 | 1123 | 2024-03-19 | 113 |**

**| 213 | 2021 | 14 | 3004 | DISC004 | 1124 | 2024-03-20 | 114 |**

**| 214 | 2022 | 15 | 3005 | DISC005 | 1125 | 2024-03-21 | 115 |**

**INNER JOIN:-**

+------+

| aid |

+------+

| 3001 |

| 3002 |

| 3003 |

| 3004 |

| 3005 |

**LEFT JOIN (or LEFT OUTER JOIN):** Returns all rows from the left table, and the matched rows from the right table. If there are no matches, NULL values are used for the columns from the right table

**SELECT columns**

**FROM table1**

**LEFT JOIN table2 ON table1.column = table2.column;**

**+------+------------+**

**| aid | password |**

**+------+------------+**

**| 3001 | password1 |**

**| 3002 | password2 |**

**| 3003 | password3 |**

**| 3004 | password4 |**

**| 3005 | password5 |**

**FULL JOIN (or FULL OUTER JOIN):** Returns all rows when there is a match in one of the tables. It combines the results of both left join and right join.

**SELECT columns**

**FROM table1**

**FULL JOIN table2 ON table1.column = table2.column;**

**+------+------------+**

**| aid | password |**

**+------+------------+**

**| 3001 | password1 |**

**| 3002 | password2 |**

**| 3003 | password3 |**

**| 3004 | password4 |**

**| 3005 | password5 |**

**| 3006 | password6 |**

**| 3007 | password7 |**

**| 3008 | password8 |**

**| 3009 | password9 |**

**| 3010 | password10 |**

**| 3011 | password11 |**

**| 3012 | password12 |**

**| 3013 | password13 |**

**| 3014 | password14 |**

**| 3015 | password15 |**

**+---------+------+------+------+----------+------------+------------+-------------+**

**| show\_id | cid | tid | aid | offer\_id | hallnumber | show\_date | seat\_number |**

**+---------+------+------+------+----------+------------+------------+-------------+**

**| 200 | 1011 | 1 | 3001 | DISC001 | 1111 | 2024-03-17 | 101 |**

**| 201 | 1012 | 2 | 3002 | DISC002 | 1112 | 2024-03-18 | 102 |**

**| 202 | 1013 | 3 | 3003 | DISC003 | 1113 | 2024-03-19 | 103 |**

**| 203 | 1014 | 4 | 3004 | DISC004 | 1114 | 2024-03-20 | 104 |**

**| 204 | 1015 | 5 | 3005 | DISC005 | 1115 | 2024-03-21 | 105 |**

**| 205 | 1016 | 6 | 3001 | DISC001 | 1116 | 2024-03-17 | 106 |**

**| 206 | 1017 | 7 | 3002 | DISC002 | 1117 | 2024-03-18 | 107 |**

**| 207 | 1018 | 8 | 3003 | DISC003 | 1118 | 2024-03-19 | 108 |**

**| 208 | 1019 | 9 | 3004 | DISC004 | 1119 | 2024-03-20 | 109 |**

**| 209 | 1020 | 10 | 3005 | DISC005 | 1120 | 2024-03-21 | 110 |**

**| 210 | 1023 | 11 | 3001 | DISC001 | 1121 | 2024-03-17 | 111 |**

**| 211 | 1024 | 12 | 3002 | DISC002 | 1122 | 2024-03-18 | 112 |**

**| 212 | 1025 | 13 | 3003 | DISC003 | 1123 | 2024-03-19 | 113 |**

**| 213 | 2021 | 14 | 3004 | DISC004 | 1124 | 2024-03-20 | 114 |**

**| 214 | 2022 | 15 | 3005 | DISC005 | 1125 | 2024-03-21 | 115 |**

**RIGHT JOIN (or RIGHT OUTER JOIN):** Returns all rows from the right table, and the matched rows from the left table. If there are no matches, NULL values are used for the columns from the left table.

**SELECT columns**

**FROM table1**

**RIGHT JOIN table2 ON table1.column = table2.column;**

**+---------+------+------+------+----------+------------+------------+-------------+**

**| show\_id | cid | tid | aid | offer\_id | hallnumber | show\_date | seat\_number |**

**+---------+------+------+------+----------+------------+------------+-------------+**

**| 200 | 1011 | 1 | 3001 | DISC001 | 1111 | 2024-03-17 | 101 |**

**| 201 | 1012 | 2 | 3002 | DISC002 | 1112 | 2024-03-18 | 102 |**

**| 202 | 1013 | 3 | 3003 | DISC003 | 1113 | 2024-03-19 | 103 |**

**| 203 | 1014 | 4 | 3004 | DISC004 | 1114 | 2024-03-20 | 104 |**

**| 204 | 1015 | 5 | 3005 | DISC005 | 1115 | 2024-03-21 | 105 |**

**| 205 | 1016 | 6 | 3001 | DISC001 | 1116 | 2024-03-17 | 106 |**

**| 206 | 1017 | 7 | 3002 | DISC002 | 1117 | 2024-03-18 | 107 |**

**| 207 | 1018 | 8 | 3003 | DISC003 | 1118 | 2024-03-19 | 108 |**

**| 208 | 1019 | 9 | 3004 | DISC004 | 1119 | 2024-03-20 | 109 |**

**| 209 | 1020 | 10 | 3005 | DISC005 | 1120 | 2024-03-21 | 110 |**

**| 210 | 1023 | 11 | 3001 | DISC001 | 1121 | 2024-03-17 | 111 |**

**| 211 | 1024 | 12 | 3002 | DISC002 | 1122 | 2024-03-18 | 112 |**

**| 212 | 1025 | 13 | 3003 | DISC003 | 1123 | 2024-03-19 | 113 |**

**| 213 | 2021 | 14 | 3004 | DISC004 | 1124 | 2024-03-20 | 114 |**

**| 214 | 2022 | 15 | 3005 | DISC005 | 1125 | 2024-03-21 | 115 |**

**VIEWS:-**

In database management systems (DBMS), a view is a virtual table that is based on the result of a SELECT query. It does not store any data of its own but instead, it presents data from one or more tables in a structured format. Views are used to simplify complex queries, hide the complexity of underlying tables, and provide a level of abstraction to users.Here are some key points about views in DBMS:1)Virtual Tables: Views are not physical tables; they are stored queries that act as virtual tables.2)Data Independence: Views provide a layer of abstraction between users and the underlying database tables. Users can interact with views without needing to know the details of the tables or the underlying data schema.3)Simplified Queries: Views allow users to write simplified queries by encapsulating complex logic or joins into a single view.4)Security: Views can be used to restrict access to certain columns or rows of a table. By granting users access to specific views instead of the underlying tables, administrators can control the data that users can see.5)Performance: While views simplify query writing, they can sometimes impact performance, especially if the underlying query is complex. However, modern database systems optimize view execution to minimize performance overhead.6)Updatability: Views can be updatable or non-updatable depending on the underlying query. Updatable views allow users to insert, update, or delete rows through the view, which can simplify data manipulation operations.7)Reuse: Views can be reused across different queries, making them a valuable tool for modularizing and organizing SQL code.

**ADMIN\_VIEW TABLE:-**

**CREATE VIEW admin\_view AS**

**SELECT aid, password**

**FROM admin;**

+------+------------+

| aid | password |

+------+------------+

| 3001 | password1 |

| 3002 | password2 |

| 3003 | password3 |

| 3004 | password4 |

| 3005 | password5 |

| 3006 | password6 |

| 3007 | password7 |

| 3008 | password8 |

| 3009 | password9 |

| 3010 | password10 |

| 3011 | password11 |

| 3012 | password12 |

| 3013 | password13 |

| 3014 | password14 |

| 3015 | password15 |

**CUSTOMER\_VIEW TABLE:-**

**CREATE VIEW customer\_view AS**

**SELECT cname, cid, password, emailid, phone\_number**

**FROM customer;**

**+--------+------+------------+--------------------+-----------------+**

**| cname | cid | password | emailid | phone\_number |**

**+--------+------+------------+--------------------+-----------------+**

**| Aarti | 1011 | password11 | aarti@example.com | +91 98765 43220 |**

**| Nitin | 1012 | password12 | nitin@example.com | +91 98765 43221 |**

**| Rita | 1013 | password13 | rita@example.com | +91 98765 43222 |**

**| Sanjay | 1014 | password14 | sanjay@example.com | +91 98765 43223 |**

**| Meena | 1015 | password15 | meena@example.com | +91 98765 43224 |**

**| Rahul | 1016 | password1 | rahul@example.com | +91 98765 43210 |**

**| Sonia | 1017 | password2 | sonia@example.com | +91 98765 43211 |**

**| Mohan | 1018 | password3 | mohan@example.com | +91 98765 43212 |**

**| Jaya | 1019 | password4 | jaya@example.com | +91 98765 43213 |**

**| Vinay | 1020 | password5 | vinay@example.com | +91 98765 43214 |**

**| Ravi | 1023 | password8 | ravi@example.com | +91 98765 43217 |**

**| Deepak | 1024 | password9 | deepak@example.com | +91 98765 43218 |**

**| Sumit | 1025 | password10 | sumit@example.com | +91 98765 43219 |**

**| Poonam | 2021 | password6 | poonam@example.com | +91 98765 43215 |**

**| Sita | 2022 | password7 | sita@example.com | +91 98765 43216 |**

**DISCOUNT\_VIEW TABLE:-**

**CREATE VIEW discount\_view AS**

**SELECT offer\_id, mname, price**

**FROM discount;**

+----------+---------------------+--------+

| offer\_id | mname | price |

+----------+---------------------+--------+

| DISC001 | Pushpa: The Rise | 150.00 |

| DISC002 | RRR | 180.00 |

| DISC003 | Acharya | 140.00 |

| DISC004 | Love Story | 120.00 |

| DISC005 | Jathi Ratnalu | 100.00 |

| DISC006 | KGF: Chapter 2 | 170.00 |

| DISC007 | Vakeel Saab | 130.00 |

| DISC008 | Bheemla Nayak | 160.00 |

| DISC009 | Radhe Shyam | 175.00 |

| DISC010 | Tuck Jagadish | 110.00 |

| DISC011 | Viraata Parvam | 145.00 |

| DISC012 | Ghani | 135.00 |

| DISC013 | Sarkaru Vaari Paata | 165.00 |

| DISC014 | Bheeshma Parvam | 125.00 |

| DISC015 | Naarappa | 155.00 |

**MOVIE\_VIEW TABLE:-**

**CREATE VIEW movie\_view AS**

**SELECT mid, mname, release\_date, duration, director, cast, budget, rating**

**FROM movie;**

**+-----+---------------------+--------------+----------+----------------------+-------------------------------------------------------------+--------+--------+**

**| mid | mname | release\_date | duration | director | cast**

**| budget | rating |**

**+-----+---------------------+--------------+----------+----------------------+-------------------------------------------------------------+--------+--------+**

**| 1 | Pushpa: The Rise | 2021-12-17 | 173 | Sukumar | Allu Arjun, Fahadh Faasil, Rashmika Mandanna | 20.00 | 8.3 |**

**| 2 | RRR | 2022-03-24 | 176 | S.S. Rajamouli | N. T. Rama Rao Jr., Ram Charan, Alia Bhatt | 35.00 | 8.7 |**

**| 3 | Acharya | 2022-04-29 | 147 | Koratala Siva | Chiranjeevi, Ram Charan, Kajal Aggarwal | 18.00 | 7.9 |**

**| 4 | Love Story | 2021-09-24 | 154 | Sekhar Kammula | Naga Chaitanya, Sai Pallavi**

**| 25.00 | 8.5 |**

**| 5 | Jathi Ratnalu | 2021-03-11 | 148 | Anudeep K.V. | Naveen Polishetty, Priyadarshi Pulikonda, Rahul Ramakrishna | 12.00 | 8.2 |**

**| 6 | KGF: Chapter 2 | 2022-04-14 | 176 | Prashanth Neel | Yash, Sanjay Dutt, Raveena Tandon | 18.00 | 8.8 |**

**| 7 | Vakeel Saab | 2021-04-09 | 154 | Venu Sriram | Pawan Kalyan, Nivetha Thomas, Anjali | 30.00 | 8.1 |**

**| 8 | Bheemla Nayak | 2022-02-25 | 158 | Sagar K Chandra | Pawan Kalyan, Rana Daggubati, Nithya Menen | 15.00 | 8.4 |**

**| 9 | Radhe Shyam | 2022-03-11 | 163 | Radha Krishna Kumar | Prabhas, Pooja Hegde | 25.00 | 7.8 |**

**| 10 | Tuck Jagadish | 2021-09-10 | 149 | Shiva Nirvana | Nani, Ritu Varma, Jagapathi Babu**

**| 40.00 | 7.7 |**

**| 11 | Viraata Parvam | 2022-04-07 | 152 | Venu Udugula | Rana Daggubati, Sai Pallavi, Priyamani | 20.00 | 8.0 |**

**| 12 | Ghani | 2022-07-15 | 160 | Kirankumar Korrapati | Varun Tej, Saiee Manjrekar, Upendra | 22.00 | 7.6 |**

**| 13 | Sarkaru Vaari Paata | 2022-08-12 | 170 | Parasuram Petla | Mahesh Babu, Keerthy Suresh**

**| 30.00 | 8.6 |**

**| 14 | Bheeshma Parvam | 2022-03-31 | 162 | Amal Neerad | Mammootty, Soubin Shahir, Sreenath Bhasi | 28.00 | 8.2 |**

**| 15 | Naarappa | 2021-07-20 | 152 | Srikanth Addala | Venkatesh Daggubati, Priyamani**

**| 22.00 | 8.0 |**

**+-----+---------------------+--------------+----------+----------------------+-------------------------------------------------------------+--------+--------+**

**SEAT\_VIEW TABLE:-**

**CREATE VIEW seat\_view AS**

**SELECT sid, sname, number\_of\_seats**

**FROM seat;**

**+-----+---------+-----------------+**

**| sid | sname | number\_of\_seats |**

**+-----+---------+-----------------+**

**| 101 | Seat 1 | 10 |**

**| 102 | Seat 2 | 20 |**

**| 103 | Seat 3 | 15 |**

**| 104 | Seat 4 | 12 |**

**| 105 | Seat 5 | 18 |**

**| 106 | Seat 6 | 25 |**

**| 107 | Seat 7 | 8 |**

**| 108 | Seat 8 | 16 |**

**| 109 | Seat 9 | 22 |**

**| 110 | Seat 10 | 14 |**

**| 111 | Seat 11 | 30 |**

**| 112 | Seat 12 | 9 |**

**| 113 | Seat 13 | 17 |**

**| 114 | Seat 14 | 21 |**

**| 115 | Seat 15 | 11 |**

**SHOW\_VIEW TABLE:-**

**CREATE VIEW shows\_view AS**

**SELECT show\_id, language, start\_time, end\_time**

**FROM shows;**

**+---------+----------+------------+----------+**

**| show\_id | language | start\_time | end\_time |**

**+---------+----------+------------+----------+**

**| 200 | Telugu | 10:00:00 | 12:00:00 |**

**| 201 | Telugu | 12:30:00 | 14:30:00 |**

**| 202 | Telugu | 15:00:00 | 17:00:00 |**

**| 203 | Telugu | 17:30:00 | 19:30:00 |**

**| 204 | Telugu | 20:00:00 | 22:00:00 |**

**| 205 | Telugu | 10:30:00 | 12:30:00 |**

**| 206 | Telugu | 13:00:00 | 15:00:00 |**

**| 207 | Telugu | 15:30:00 | 17:30:00 |**

**| 208 | Telugu | 18:00:00 | 20:00:00 |**

**| 209 | Telugu | 20:30:00 | 22:30:00 |**

**| 210 | Telugu | 11:00:00 | 13:00:00 |**

**| 211 | Telugu | 13:30:00 | 15:30:00 |**

**| 212 | Telugu | 16:00:00 | 18:00:00 |**

**| 213 | Telugu | 18:30:00 | 20:30:00 |**

**| 214 | Telugu | 21:00:00 | 23:00:00 |**

**THEATRE\_VIEW TABLE:-**

**CREATE VIEW theatre\_view AS**

**SELECT tid, tname, location**

**FROM theatre;**

**+-----+-----------------------------------------+---------------+**

**| tid | tname | location |**

**+-----+-----------------------------------------+---------------+**

**| 1 | PVR Cinemas - Banjara Hills | Banjara Hills |**

**| 2 | INOX - GVK One Mall | Jubilee Hills |**

**| 3 | Cinepolis - Manjeera Mall | Kukatpally |**

**| 4 | IMAX - Prasads | Tank Bund |**

**| 5 | Asian Cinemas - Gachibowli | Gachibowli |**

**| 6 | Mukta A2 Cinemas - Rashtrapati Road | Secunderabad |**

**| 7 | AMB Cinemas - Gachibowli | Gachibowli |**

**| 8 | Cineplex - Hyderabad Central Mall | Panjagutta |**

**| 9 | Miraj Cinemas - Shalini Shivani Complex | Abids |**

**| 10 | Carnival Cinemas - Ameerpet | Ameerpet |**

**| 11 | CineTown - Miyapur | Miyapur |**

**| 12 | INOX - Maheshwari Parmeshwari Mall | Kachiguda |**

**| 13 | Asian Ganga - Attapur | Attapur |**

**| 14 | BIG Cinemas - Tivoli Cinema | Tivoli Circle |**

**| 15 | Asian Cine Square Multiplex - Uppal | Uppal |**

**TICKET\_VIEW TABLE:-**

**CREATE VIEW ticket\_view AS**

**SELECT show\_id, cid, tid, aid, offer\_id, hallnumber, show\_date, seat\_number**

**FROM ticket;**

**+---------+------+------+------+----------+------------+------------+-------------+**

**| show\_id | cid | tid | aid | offer\_id | hallnumber | show\_date | seat\_number |**

**+---------+------+------+------+----------+------------+------------+-------------+**

**| 200 | 1011 | 1 | 3001 | DISC001 | 1111 | 2024-03-17 | 101 |**

**| 201 | 1012 | 2 | 3002 | DISC002 | 1112 | 2024-03-18 | 102 |**

**| 202 | 1013 | 3 | 3003 | DISC003 | 1113 | 2024-03-19 | 103 |**

**| 203 | 1014 | 4 | 3004 | DISC004 | 1114 | 2024-03-20 | 104 |**

**| 204 | 1015 | 5 | 3005 | DISC005 | 1115 | 2024-03-21 | 105 |**

**| 205 | 1016 | 6 | 3001 | DISC001 | 1116 | 2024-03-17 | 106 |**

**| 206 | 1017 | 7 | 3002 | DISC002 | 1117 | 2024-03-18 | 107 |**

**| 207 | 1018 | 8 | 3003 | DISC003 | 1118 | 2024-03-19 | 108 |**

**| 208 | 1019 | 9 | 3004 | DISC004 | 1119 | 2024-03-20 | 109 |**

**| 209 | 1020 | 10 | 3005 | DISC005 | 1120 | 2024-03-21 | 110 |**

**| 210 | 1023 | 11 | 3001 | DISC001 | 1121 | 2024-03-17 | 111 |**

**| 211 | 1024 | 12 | 3002 | DISC002 | 1122 | 2024-03-18 | 112 |**

**| 212 | 1025 | 13 | 3003 | DISC003 | 1123 | 2024-03-19 | 113 |**

**| 213 | 2021 | 14 | 3004 | DISC004 | 1124 | 2024-03-20 | 114 |**

**| 214 | 2022 | 15 | 3005 | DISC005 | 1125 | 2024-03-21 | 115 |**

**SUBQUERIES:-**

A subquery, also known as a nested query or inner query, is a query nested within another SQL statement, such as SELECT, INSERT, UPDATE, or DELETE. Subqueries are enclosed within parentheses and can be used in various parts of a SQL statement, including WHERE clauses, FROM clauses, HAVING clauses, and more. Subqueries allow you to perform operations or retrieve data based on the results of another query.

Here's the basic syntax of a subquery:

**SELECT column1, column2, ...**

**FROM table**

**WHERE column1 = (SELECT column1 FROM another\_table WHERE condition);**

Now, let's explore different types of subqueries:

**Single-row Subquery**: A subquery that returns only one row and one column.

**SELECT column1**

**FROM table1**

**WHERE column1 = (SELECT column1 FROM table2 WHERE condition);**

**Eg:-**

SELECT emailid

FROM admin

WHERE aid = (SELECT MAX(aid) FROM admin);

**Multi-row Subquery**: A subquery that returns multiple rows but only one column.

**SELECT column1**

**FROM table1**

**WHERE column1 IN (SELECT column1 FROM table2 WHERE condition);**

**Eg:-**

SELECT emailid

FROM customer

WHERE password LIKE 'password1%';

**Multi-column Subquery**: A subquery that returns multiple rows and columns.

**SELECT column1, column2**

**FROM table1**

**WHERE (column1, column2) IN (SELECT column1, column2 FROM table2 WHERE condition);**

**Eg:-**

SELECT name, emailid

FROM admin

WHERE (aid, password) IN (SELECT cid, password FROM customer);

**Correlated Subquery:** A subquery that references columns from the outer query, allowing you to filter results based on values in the outer query.

**SELECT column1**

**FROM table1 t1**

**WHERE column1 > (SELECT AVG(column1) FROM table1 t2 WHERE t2.column2 = t1.column2);**

**Eg:-**

SELECT name,

(SELECT COUNT(\*) FROM customer WHERE admin.aid = customer.aid) AS customer\_count

FROM admin;

**Scalar Subquery:**A scalar subquery returns a single value (single row, single column).

**SELECT column1, (SELECT MAX(column2) FROM table2) AS max\_column2**

**FROM table1**;

**Eg:-**

SELECT MAX(LENGTH(password)) AS max\_password\_length

FROM customer;