

SQL QUERIES

MySQL Workbench Local instance MySQL80

DB Bank*

```

2 /* ANALYSIS ON BANK DATABASE */
3
4 • use bank;           #to use the database bank
5 • show tables;        #to view the tables under the database bank
6
7 /*1.What is the total balance amount held by each customer across all account types?*/
8 • select bd.customer_id, bc.customer_name, sum(bd.Balance_amount) as Total_Balance
9   from bank_account_details bd
10  join bank_customer bc on bd.customer_id = bc.customer_id
11  group by bd.customer_id, bc.customer_name
12  order by Total_Balance desc;
13
14 /*Analysis:
15 This query aggregates the balance amount across all accounts a customer holds,
16 giving a total sum per customer.
17
18
19 /*2..which customers have inactive accounts, and what type of accounts are they?*/
20 • select bd.customer_id, bc.customer_name, bd.Account_Number, bd.account_type
21   from bank_account_details bd join bank_customer bc on bd.customer_id = bc.customer_id
22   where Account_Status='inactive';
23
24 /*Analysis:
25 This query helps the bank track inactive accounts,
26 which may require re-engagement efforts or regulatory checks.
27 Identifying the type of inactive accounts (Savings, Recurring Deposits, Credit Cards)
28 can assist in strategizing personalized offers or reminders*/
29
30
31
32

```

Result Grid

customer_id	customer_name	Total_Balance
123001	Oliver	960000
123004	Jack	819500
123002	George	790000
123007	Charlie	738000
123003	Harry	75000
123006	Noah	455000
123005	Jacob	300000
123008	Robin	0

Result 82

Output

Action Output

Result Grid

customer_id	customer_name	Account_Number	account_type
123003	Harry	4000-1956-2900	SAVINGS
123008	Robin	5000-1700-7755	SAVINGS
123007	Charlie	9000-1700-7777-4321	Credit Card

Result 83

Output

Action Output

Result Grid

province	total_transaction
MN	176900.00
NY	106500.00
CA	-11000.00

Result 84

Output

Action Output

MySQL Workbench Local instance MySQL80

```

44
45
46 /*4.Which customers have more than one account, and what types of accounts do they hold?*/
47 • select bd.customer_id, bc.customer_name, group_concat(distinct bd.account_type),count(bd.Account_Number) as totalnum_of_accounts
48 from bank_account_details bd join bank_customer bc on bd.customer_id = bc.customer_id
49 group by bd.customer_id, bc.customer_name
50 having totalnum_of_accounts > 1
51 order by totalnum_of_accounts desc;
52
53 /*Analysis:
54 This query identifies customers with multiple accounts and lists the types of accounts they hold.
55 It helps in understanding customer engagement with the bank and in targeting them for specialized financial products.*/
56
57
58

```

Result Grid | Filter Rows: Export: Wrap Cell Content: Result Grid

customer_id	customer_name	group_concat(distinct bd.account_type)	totalnum_of_accounts
123007	Charlie	Add-on Credit Card,Credit Card,RECURRING D...	4
123004	Jack	RECURRING DEPOSITS,SAVINGS	3
123006	Noah	Add-on Credit Card,Credit Card,SAVINGS	3
123001	Oliver	RECURRING DEPOSITS,SAVINGS	2
123002	George	RECURRING DEPOSITS,SAVINGS	2

Result 85 x

Action Output

Object Info Session

MySQL Workbench Local instance MySQL80

```

57
58
59 /*5. What is the average transaction amount for each type of transaction channel?*/
60 • select Transaction_channel,abs(round(avg(transaction_amount),2))as avg_transaction_amount ,count(*) transactions_count
61 from bank_account_transaction
62 group by Transaction_channel
63 order by avg_transaction_amount desc;
64
65 /*Analysis:
66 This query helps in analyzing the average amount transacted through different channels (ATM, POS, Net Banking, etc.).
67 It can be useful for identifying preferred transaction methods and optimizing banking services*/
68
69
70

```

Result Grid | Filter Rows: Export: Wrap Cell Content: Result Grid

Transaction_channel	avg_transaction_amount	transactions_count
ECS transfer	38625.00	8
cheque deposit	23000.00	1
Shopping Cart	7166.67	3
ATM withdrawal	6500.00	1
POS-Walmart	6000.00	3
Bankers cheque	6000.00	1
Net banking	3000.00	1
ATM withdrawal	2000.00	1
UPI transfer	1600.00	1
Cash Deposit	500.00	2

Result 92 x

Action Output

Object Info Session

MySQL Workbench Local instance MySQL80

```

69
70
71 /*6.Which customers have savings accounts with a balance of more than 500,000?*/
72 • select bd.customer_id, bc.customer_name, bd.account_Number, bd.balance_amount
73 from bank_account_details bd join bank_customer bc on bd.customer_id = bc.customer_id
74 where bd.Account_type = 'savings' AND bd.Balance_amount > 500000
75 order by bd.Balance_amount desc;
76
77 /*Analysis:
78 This query helps in identifying high-value savings account holders.
79 These customers can be targeted for premium banking services, investment opportunities, or loyalty programs*/
80
81
82

```

Result Grid | Filter Rows: Export: Wrap Cell Content: Result Grid

customer_id	customer_name	account_Number	balance_amount
123003	Harry	4000-1956-2900	750000
123004	Jack	4000-1956-3401	650000

Result 93 x

Action Output

Object Info Session

MySQL Workbench Local instance MySQL80

DB Bank*

```

83
84
85 /*7.Which customers have made the most transactions, and how much have they transacted in total?*/
86 • select bd.customer_id, bc.customer_name, count(bt.Account_Number) as Transaction_Count,abs(sum(bt.Transaction_amount))as Total_Transaction
87 from bank_account_transaction bt join bank_account_details bd on bt.Account_Number = bd.Account_Number
88 join bank_customer bc on bd.customer_id = bc.customer_id
89 group by bd.customer_id, bc.customer_name
90 order by Transaction_Count desc;
91
92 /*Analysis:
93 This query helps identify highly active customers based on their transaction count and total transaction amount.
94 The bank can use this data for customer relationship management, fraud monitoring, or offering tailored financial products*/
95
96

```

Result Grid

customer_id	customer_name	Transaction_Count	Total_Transaction
123007	Charlie	9	213500.00
123002	George	5	8400.00
123004	Jack	3	88000.00
123006	Noah	3	29000.00
123001	Oliver	1	2000.00
123005	Jacob	1	6500.00

Result 95

Action Output

Object Info **Session**

MySQL Workbench Local instance MySQL80

DB Bank*

```

96
97
98 /*8.what is the total balance amount held by customers in each state?*/
99 • select bc.state_code, sum(bd.Balance_amount) as Total_Balance
100 from bank_account_details bd join bank_customer bc on bd.customer_id = bc.customer_id
101 group by bc.state_code
102 order by Total_Balance desc;
103
104 /*Analysis:
105 This query provides insights into the distribution of bank funds across different states.
106 It helps in regional financial planning, targeted marketing, and assessing state-wise banking performance*/
107
108
109

```

Result Grid

state_code	Total_Balance
CA	18545000
MN	15735000
SFO	300000
NY	0

Result 96

Action Output

Object Info **Session**

MySQL Workbench Local instance MySQL80

DB Bank*

```

110 /*9.Print credit card transactions with the sum of transaction_amount on all Fridays and the sum of transaction_amount on all other days.*/
111 • select
112 case
113 when dayname(Transaction_Date) = 'Friday' then 'Friday' else 'Other Days'
114 end as Day_Type,
115 abs(sum(Transaction_amount)) as Total_Transaction_Amount
116 from bank_account_transaction bt join bank_account_details bd on bt.Account_Number = bd.Account_Number
117 where bd.Account_type in ('Credit Card', 'Add-on Credit Card')
118 group by Day_Type;
119
120 /*Analysis:
121 If the Friday total is significantly higher, it may indicate that customers prefer using credit cards for weekend purchases, shopping, or entertainment.
122 If the other days total is higher, it suggests more distributed spending throughout the week*/
123

```

Result Grid

Day_Type	Total_Transaction_Amount
Other Days	27500.00
Friday	8000.00

Result 98

Action Output

Object Info **Session**

MySQL Workbench Local instance MySQL80

```

125  /*9a.Further analysis on Transaction Channel,where those credit card transactions done on fridays and other days*/
126  select Transaction_channel,
127  case
128  when dayname(Transaction_Date) = 'Friday' then 'Friday' else 'Other Days'
129  end as Day_Type,
130  abs(sum(Transaction_amount)) as Total_Transaction_Amount
131  from bank_account_transaction bt join bank_account_details bd on bt.Account_Number = bd.Account_Number
132  where bd.Account_type in ('Credit Card', 'Add-on Credit Card')
133  group by Transaction_channel,Day_Type;
134
135  /*Analysis:
136  This helps determine where customers are using their credit cards the most
137  i.e.in which transaction channel(e.g., Online Shopping, POS, ATMs, etc.)*/
138

```

Table: bank_account_details

Customer_id	Account_Number	Account_type	Balance_amount	Account_status	Relationship_type

Result Grid

Transaction_Channel	Day_Type	Total_Transaction_Amount
POS-Walmart	Other Days	14000.00
Shopping Cart	Other Days	13500.00
Shopping Cart	Friday	8000.00

MySQL Workbench Local instance MySQL80

```

142  /*10.Show the details of credit cards along with the aggregate transaction amount during holidays and non-holidays.*/
143  select
144  case
145  when bt.Transaction_Date in (select Holiday from bank_holidays) then 'Holiday' else 'Non-Holiday'
146  end as Day_Type,
147  bd.customer_id,bc.customer_name,bd.Account_Number,bd.Account_type,abs(sum(Transaction_amount))as Total_Transaction_Amount
148  from bank_account_transaction bt join bank_account_details bd on bt.Account_Number = bd.Account_Number
149  join bank_customer bc ON bd.customer_id = bc.customer_id
150  where bd.Account_type in ('Credit Card', 'Add-on Credit Card')
151  group by Day_Type, bd.customer_id, bc.customer_name, bd.Account_Number, bd.Account_type;
152
153  /*Analysis:
154  If holiday transactions are significantly higher, it indicates that customers use credit cards more for holiday shopping, travel, or dining.
155  If non-holiday transactions dominate, spending is more routine, suggesting no major seasonal impact.*/
156

```

Table: bank_account_details

Customer_id	Account_Number	Account_type	Balance_amount	Account_status	Relationship_type

Result Grid

Day_Type	customer_id	customer_name	Account_Number	Account_type	Total_Transaction_Amount
Non-Holiday	123006	Noah	5800-1700-9800-7755	Credit Card	9000.00
Non-Holiday	123006	Noah	5890-1700-7706-8912	Add-on Credit Card	11000.00
Non-Holiday	123007	Charlie	9000-1700-7777-4321	Credit Card	7500.00
Holiday	123007	Charlie	9000-1700-7777-4321	Credit Card	8000.00

MySQL Workbench Local instance MySQL80

```

158
159
160
161  /*11.customers who done transaction in the holiday month of march and send a holiday greeting message*/
162  select t1.*,t3.Account_Number, t2.account_type, t3.Transaction_Date,'HAPPY HOLIDAY'AS Holiday_Greeting
163  from bank_customer t1 join bank_account_details t2 on t1.customer_id=t2.Customer_id
164  join bank_account_transaction t3 on t2.account_number = t3.account_number
165  where t3.Transaction_Date in (select HOLIDAY from BANK_HOLIDAYS where month(HOLIDAY)=3);
166
167  /*Analysis:
168  Identifying customers who made transactions on bank holidays in March.
169  Send them a personalized greeting message: "Happy Holiday" as a part of customer engagement.*/
170
171

```

Table: bank_account_details

Customer_id	Account_Number	Account_type	Balance_amount	Account_status	Relationship_type

Result Grid

customer_id	customer_name	Address	state_code	Telephone	Account_Number	account_type	Transaction_Date	Holiday_Greeting
123007	Charlie	125-1,Richfield	MN	1897617666	9000-1700-7777-4321	Credit Card	2020-03-13	HAPPY HOLIDAY

MySQL Workbench Local instance MySQL80

```

170
171
172 /*12.What is the total interest accrued for each account type?*/
173 • select bd.Account_type,abs(sum(bd.Balance_amount * bi.interest_rate)) as Total_Interest_Accrued
174   from bank_account_details bd
175   join bank_interest_rate bi on bd.Account_type = bi.account_type
176   group by bd.Account_type
177   order by Total_Interest_Accrued desc;
178
179 /*Analysis:
180 This query helps the bank understand which account types generate the most interest liabilities.*/
181
182
183

```

Result Grid | Filter Rows: Export: Wrap Cell Content: Result Grid

Account_Type	Total_Interest_Accrued
RECURRING DEPOSITS	2202550.00
SAVINGS	124600.00

Result 103 x Output Action Output # Time Action Metadata Duration / Fetch Object Info Session Read Only

MySQL Workbench Local instance MySQL80

```

184
185 /*13. What is the interest accrued on the highest and lowest balance accounts*/
186 • (select bd.customer_id,bc.customer_name,bd.Account_Number,bd.Balance_amount,bi.interest_rate,
187   ROUND((bd.Balance_amount * bi.interest_rate), 2) as Accrued_Interest,"highest_balance_account"as balance
188   from bank_account_details bd
189   join bank_customer bc on bd.customer_id = bc.customer_id
190   join bank_interest_rate bi on bd.Account_type = bi.account_type
191   order by bd.Balance_amount desc
192   limit 1)
193 union all
194 (select bd.customer_id,bc.customer_name,bd.Account_Number,bd.Balance_amount,bi.interest_rate,
195   ROUND((bd.Balance_amount * bi.interest_rate), 2) as Accrued_Interest,"lowest_balance_account"as balance
196   from bank_account_details bd
197   join bank_customer bc on bd.customer_id = bc.customer_id
198   join bank_interest_rate bi on bd.Account_type = bi.account_type
199   order by bd.Balance_amount asc
200   limit 1);
201
202 /*Analysis:
203 Helps track customers with extreme balances and evaluate their interest earnings for banking strategies.

```

Result Grid | Filter Rows: Export: Wrap Cell Content: Result Grid

customer_id	customer_name	Account_Number	Balance_amount	interest_rate	Accrued_Interest	balance
123001	Oliver	5000-1700-3456	9400000	0.07	658000.00	highest_balance_account
123008	Robin	5000-1700-7755	0	0.04	0.00	lowest_balance_account

Result 104 x Output Action Output # Time Action Metadata Duration / Fetch Object Info Session Read Only

MySQL Workbench Local instance MySQL80

```

204
205
206 /*14.Find customers whose total interest earnings exceed 50,000.*/
207 • select bd.customer_id,bc.customer_name,sum(bd.Balance_amount * bi.interest_rate) as Total_Interest
208   from bank_account_details bd
209   join bank_customer bc on bd.customer_id = bc.customer_id
210   join bank_interest_rate bi on bd.Account_type = bi.account_type
211   group by bd.customer_id, bc.customer_name
212   having Total_Interest > 50000
213   order by Total_Interest desc;
214
215 /*Analysis:
216 Identifies high-interest-earning customers who may need premium banking services or targeted investment plans*/
217

```

Result Grid | Filter Rows: Export: Wrap Cell Content: Result Grid

customer_id	customer_name	Total_Interest
123001	Oliver	666000.00
123004	Jack	554000.00
123002	George	541000.00
123007	Charlie	505950.00

Result 105 x Output Action Output # Time Action Metadata Duration / Fetch Object Info Session Read Only

MySQL Workbench

Local instance MySQL80 X

File Edit View Query Database Server Tools Scripting Help

Navigator DB Bank*

SCHEMAS Tables

Table: bank_account_details

Columns:

- Customer_id int
- Account_Number varchar(19) PK
- Account_type varchar(25)
- Balance_amount int
- Account_status varchar(10)
- Relationship_type varchar(1)

Information

DB Bank*

```

220 /*15.What is the difference between interest payable and interest receivable for the bank?*/
221 select sum(case
222     when bd.Account_type in ('savings', 'recurring deposits') then bd.Balance_amount * bi.interest_rate else 0
223     END) as Total_Interest_Payable,
224     sum(case
225         when bd.Account_type in ('credit card', 'Loan') then bd.Balance_amount * bi.interest_rate else 0
226         END) as Total_Interest_Receivable,
227     sum((case
228         when bd.Account_type in ('Credit Card', 'Loan') then bd.Balance_amount * bi.interest_rate else 0
229         end) -
230     (case
231         when bd.Account_type in ('savings', 'recurring deposits') then bd.Balance_amount * bi.interest_rate else 0
232         end)
233     ) as Net_Interest_Revenue
234     from bank_account_details bd
235     join bank_interest_rate bi on bd.Account_type = bi.account_type;
236
237 /*Analysis:
238 Calculates the bank's net revenue from interest
239 by comparing interest payable (on deposits) vs. interest receivable (on loans & credit cards)*/

```

Result Grid | Filter Rows: Export: Wrap Cell Content: Result Grid

Total_Interest_Payable	Total_Interest_Receivable	Net_Interest_Revenue
2327150.00	0.00	-2327150.00

Result 106 X

Output Action Output # Time Action Metrics Duration / Fetch

Object Info Session Read Only