

ASSIGNMENT

SUBMITTED BY: Fahimunnisha A

TOPIC: Banking System

Tasks 1: Database Design:

1. Create the database named "HMBank"

```
create database HMBANK;  
use HMBANK;
```

2. Define the schema for the Customers, Accounts, and Transactions tables based on the provided schema.

Customers Table:

Column Name	Data Type	Constraints
CustomerID	INT	PRIMARY KEY, AUTO_INCREMENT
FirstName	VARCHAR(50)	NOT NULL
LastName	VARCHAR(50)	NOT NULL
DateOfBirth	DATE	NOT NULL
Email	VARCHAR(100)	UNIQUE
Phone	VARCHAR(15)	
Address	VARCHAR(255)	

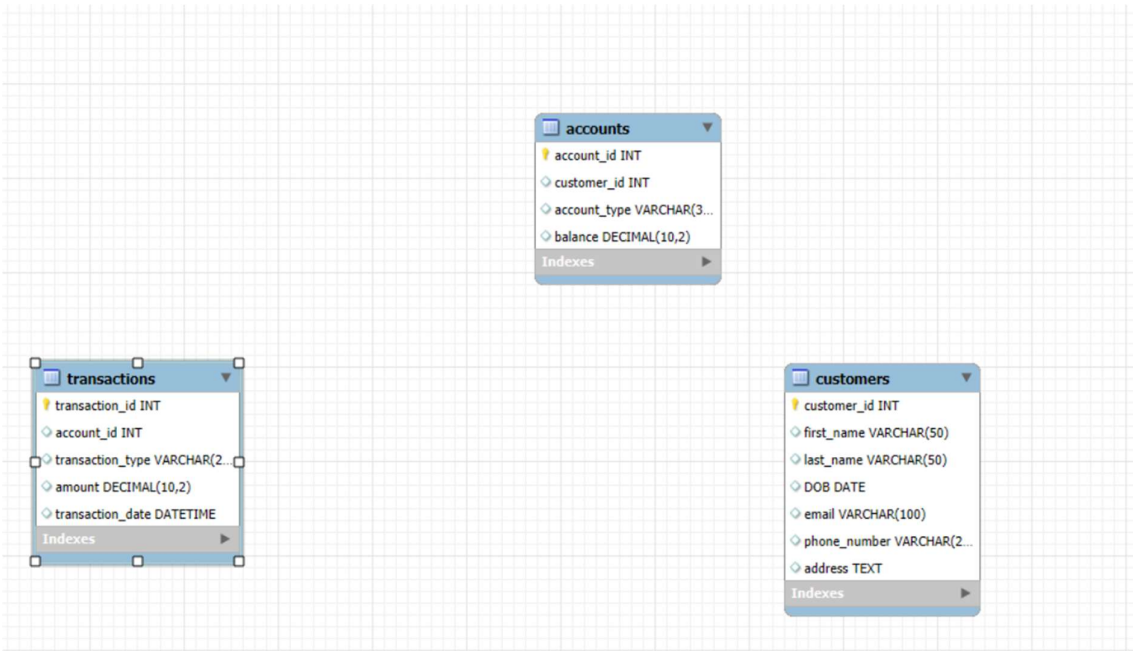
Accounts Table:

Column Name	Data type	Constraints
AccountID	INT	PRIMARY KEY, AUTO_INCREMENT
CustomerID	INT	FOREIGN KEY → Customers(CustomerID)
AccountType	VARCHAR(20)	NOT NULL
Balance	DECIMAL(12,2)	DEFAULT 0.00
OpenDate	DATE	NOT NULL

Transactions table:

Column Name	Data type	Constraints
TransactionID	INT	PRIMARY KEY, AUTO_INCREMENT
AccountID	INT	FOREIGN KEY → Accounts(AccountID)
TransactionType	VARCHAR(20)	NOT NULL (Deposit, Withdraw, etc.)
Amount	DECIMAL(12, 2)	NOT NULL
TransactionDate	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP

3. Create an ERD (Entity Relationship Diagram) for the database.



4. Create appropriate Primary Key and Foreign Key constraints for referential integrity.

```
CREATE TABLE Transactions (  
    transaction_id INT PRIMARY KEY,  
    account_id INT,  
    transaction_type VARCHAR(20),  
    amount DECIMAL(10,2),  
    transaction_date DATETIME,  
    FOREIGN KEY (account_id) REFERENCES Accounts(account_id)  
);
```

5. Write SQL scripts to create the mentioned tables with appropriate data types, constraints, and relationships.

☐ Customers

☐ Accounts

☐ Transactions

```
CREATE TABLE Customers (  
    customer_id INT PRIMARY KEY,  
    first_name VARCHAR(50),  
    last_name VARCHAR(50),  
    DOB DATE,  
    email VARCHAR(100),  
    phone_number VARCHAR(20),  
    address TEXT  
);
```

```
) CREATE TABLE Accounts (  
    account_id INT PRIMARY KEY,  
    customer_id INT,  
    account_type VARCHAR(30),  
    balance DECIMAL(10,2),  
    FOREIGN KEY (customer_id) REFERENCES Customers(customer_id)  
);
```

```
CREATE TABLE Transactions (  
    transaction_id INT PRIMARY KEY,  
    account_id INT,  
    transaction_type VARCHAR(20),  
    amount DECIMAL(10,2),  
    transaction_date DATETIME,  
    FOREIGN KEY (account_id) REFERENCES Accounts(account_id)  
);
```

Tasks 2: Select, Where, Between, AND, LIKE:

1. Insert at least 10 sample records into each of the following tables.

- ☐ Customers
- ☐ Accounts
- ☐ Transactions

12 • INSERT INTO Customers VALUES

```

13 (1, 'John', 'Doe', '1985-06-15', 'john.doe@example.com', '98765432
14 (2, 'Jane', 'Smith', '1990-11-20', 'jane.smith@example.com', '9876
15 (3, 'Alice', 'Johnson', '1975-09-30', 'alice.j@example.com', '9876
16 (4, 'Bob', 'Brown', '2000-01-01', 'bob.b@example.com', '9876543213
17 (5, 'Charlie', 'Davis', '1995-03-12', 'charlie.d@example.com', '98
18 (6, 'Emily', 'Clark', '1988-07-08', 'emily.c@example.com', '987654
19 (7, 'Frank', 'Wright', '1992-05-05', 'frank.w@example.com', '9876543216
20 (8, 'Grace', 'Lee', '1983-12-25', 'grace.l@example.com', '9876543217
21 (9, 'Hank', 'Green', '1970-10-10', 'hank.g@example.com', '9876543218
22 (10, 'Ivy', 'King', '1998-04-18', 'ivy.k@example.com', '9876543219
23 (11, 'Judy', 'White', '1960-02-01', 'judy.w@example.com', '9876543220
24 (12, 'Eugene', 'Black', '1973-08-14', 'eugene.b@example.com', '9876543221
25 (13, 'Sally', 'Hill', '1955-05-27', 'sally.h@example.com', '9876543222
26 (14, 'Timothy', 'Scott', '1968-09-09', 'timothy.s@example.com', '9876543223
27 (15, 'Lillian', 'Young', '1942-03-22', 'lillian.y@example.com', '9876543224
28 (16, 'Harold', 'Adams', '1958-11-11', 'harold.a@example.com', '9876543225
29 (17, 'Betty', 'Baker', '1965-07-04', 'betty.b@example.com', '9876543226
30 (18, 'George', 'Nelson', '1979-04-25', 'george.n@example.com', '9876543227
31 (19, 'Dorothy', 'Carter', '1981-12-18', 'dorothy.c@example.com', '9876543228
32 (20, 'Kenneth', 'Mitchell', '1989-06-03', 'kenneth.m@example.com', '9876543229
33 (21, 'Shirley', 'Perez', '1993-01-20', 'shirley.p@example.com', '9876543230
34 (22, 'Roy', 'Roberts', '1997-08-07', 'roy.r@example.com', '9876543231
35 (23, 'Linda', 'Garcia', '2001-03-15', 'linda.g@example.com', '9876543232
36 (24, 'William', 'Martinez', '2004-10-02', 'william.m@example.com', '9876543233
37 (25, 'Elizabeth', 'Hernandez', '2007-05-19', 'elizabeth.h@example.com', '9876543234
38 (26, 'Christopher', 'Lopez', '2010-12-06', 'christopher.l@example.com', '9876543235
39 (27, 'Margaret', 'Gonzalez', '2013-09-23', 'margaret.g@example.com', '9876543236
40 (28, 'Daniel', 'Wilson', '2016-04-10', 'daniel.w@example.com', '9876543237
41 (29, 'Patricia', 'Anderson', '2019-01-27', 'patricia.a@example.com', '9876543238
42 (30, 'Joseph', 'Taylor', '2022-06-14', 'joseph.t@example.com', '9876543239
43 (31, 'Susan', 'Thomas', '2025-03-01', 'susan.t@example.com', '9876543240
44 (32, 'Robert', 'Jackson', '2028-08-18', 'robert.j@example.com', '9876543241
45 (33, 'Katherine', 'White', '2031-05-05', 'katherine.w@example.com', '9876543242
46 (34, 'Steven', 'Harris', '2034-10-22', 'steven.h@example.com', '9876543243
47 (35, 'Amanda', 'Clark', '2037-07-09', 'amanda.c@example.com', '9876543244
48 (36, 'Matthew', 'Lewis', '2040-02-26', 'matthew.l@example.com', '9876543245
49 (37, 'Deborah', 'Walker', '2043-09-13', 'deborah.w@example.com', '9876543246
50 (38, 'Anthony', 'Hall', '2046-04-01', 'anthony.h@example.com', '9876543247
51 (39, 'Michelle', 'Young', '2049-11-18', 'michelle.y@example.com', '9876543248
52 (40, 'Christopher', 'King', '2052-08-05', 'christopher.k@example.com', '9876543249
53 (41, 'Dorothy', 'Wright', '2055-05-22', 'dorothy.w@example.com', '9876543250
54 (42, 'Kenneth', 'Green', '2058-02-09', 'kenneth.g@example.com', '9876543251
55 (43, 'Shirley', 'Nelson', '2061-09-26', 'shirley.n@example.com', '9876543252
56 (44, 'Roy', 'Carter', '2064-06-13', 'roy.c@example.com', '9876543253
57 (45, 'Linda', 'Mitchell', '2067-03-01', 'linda.m@example.com', '9876543254
58 (46, 'William', 'Perez', '2070-10-18', 'william.p@example.com', '9876543255
59 (47, 'Elizabeth', 'Roberts', '2073-07-05', 'elizabeth.r@example.com', '9876543256
60 (48, 'Christopher', 'Garcia', '2076-04-22', 'christopher.g@example.com', '9876543257
61 (49, 'Margaret', 'Martinez', '2079-01-09', 'margaret.m@example.com', '9876543258
62 (50, 'Daniel', 'Hernandez', '2082-08-26', 'daniel.h@example.com', '9876543259
63 (51, 'Patricia', 'Lopez', '2085-05-13', 'patricia.l@example.com', '9876543260
64 (52, 'Joseph', 'Gonzalez', '2088-02-01', 'joseph.g@example.com', '9876543261
65 (53, 'Susan', 'Wilson', '2091-09-18', 'susan.w@example.com', '9876543262
66 (54, 'Robert', 'Anderson', '2094-06-05', 'robert.a@example.com', '9876543263
67 (55, 'Katherine', 'Taylor', '2097-03-22', 'katherine.t@example.com', '9876543264
68 (56, 'Steven', 'Thomas', '2100-12-09', 'steven.t@example.com', '9876543265
69 (57, 'Amanda', 'Jackson', '2103-09-26', 'amanda.j@example.com', '9876543266
70 (58, 'Matthew', 'White', '2106-06-13', 'matthew.w@example.com', '9876543267
71 (59, 'Deborah', 'Harris', '2109-03-01', 'deborah.h@example.com', '9876543268
72 (60, 'Anthony', 'Clark', '2112-10-18', 'anthony.c@example.com', '9876543269
73 (61, 'Michelle', 'Lewis', '2115-07-05', 'michelle.l@example.com', '9876543270
74 (62, 'Christopher', 'Walker', '2118-04-22', 'christopher.w@example.com', '9876543271
75 (63, 'Dorothy', 'Hall', '2121-01-09', 'dorothy.h@example.com', '9876543272
76 (64, 'Kenneth', 'Young', '2124-08-26', 'kenneth.y@example.com', '9876543273
77 (65, 'Shirley', 'King', '2127-05-13', 'shirley.k@example.com', '9876543274
78 (66, 'Roy', 'Wright', '2130-02-01', 'roy.w@example.com', '9876543275
79 (67, 'Linda', 'Green', '2133-09-18', 'linda.g@example.com', '9876543276
80 (68, 'William', 'Nelson', '2136-06-05', 'william.n@example.com', '9876543277
81 (69, 'Elizabeth', 'Carter', '2139-03-22', 'elizabeth.c@example.com', '9876543278
82 (70, 'Christopher', 'Mitchell', '2142-12-09', 'christopher.m@example.com', '9876543279
83 (71, 'Margaret', 'Perez', '2145-09-26', 'margaret.p@example.com', '9876543280
84 (72, 'Daniel', 'Roberts', '2148-06-13', 'daniel.r@example.com', '9876543281
85 (73, 'Patricia', 'Garcia', '2151-03-01', 'patricia.g@example.com', '9876543282
86 (74, 'Joseph', 'Martinez', '2154-10-18', 'joseph.m@example.com', '9876543283
87 (75, 'Susan', 'Hernandez', '2157-07-05', 'susan.h@example.com', '9876543284
88 (76, 'Robert', 'Lopez', '2160-04-22', 'robert.l@example.com', '9876543285
89 (77, 'Katherine', 'Gonzalez', '2163-01-09', 'katherine.g@example.com', '9876543286
90 (78, 'Steven', 'Wilson', '2166-08-26', 'steven.w@example.com', '9876543287
91 (79, 'Amanda', 'Anderson', '2169-05-13', 'amanda.a@example.com', '9876543288
92 (80, 'Matthew', 'Taylor', '2172-02-01', 'matthew.t@example.com', '9876543289
93 (81, 'Deborah', 'Thomas', '2175-09-18', 'deborah.t@example.com', '9876543290
94 (82, 'Anthony', 'Jackson', '2178-06-05', 'anthony.j@example.com', '9876543291
95 (83, 'Michelle', 'White', '2181-03-22', 'michelle.w@example.com', '9876543292
96 (84, 'Christopher', 'Harris', '2184-12-09', 'christopher.h@example.com', '9876543293
97 (85, 'Dorothy', 'Clark', '2187-09-26', 'dorothy.c@example.com', '9876543294
98 (86, 'Kenneth', 'Lewis', '2190-06-13', 'kenneth.l@example.com', '9876543295
99 (87, 'Shirley', 'Walker', '2193-03-01', 'shirley.w@example.com', '9876543296
100 (88, 'Roy', 'Hall', '2196-10-18', 'roy.h@example.com', '9876543297
101 (89, 'Linda', 'Young', '2199-07-05', 'linda.y@example.com', '9876543298
102 (90, 'William', 'King', '2202-04-22', 'william.k@example.com', '9876543299
103 (91, 'Elizabeth', 'Wright', '2205-01-09', 'elizabeth.w@example.com', '9876543300
104 (92, 'Christopher', 'Green', '2208-08-26', 'christopher.g@example.com', '9876543301
105 (93, 'Margaret', 'Nelson', '2211-05-13', 'margaret.n@example.com', '9876543302
106 (94, 'Daniel', 'Carter', '2214-02-01', 'daniel.c@example.com', '9876543303
107 (95, 'Patricia', 'Mitchell', '2217-09-18', 'patricia.m@example.com', '9876543304
108 (96, 'Joseph', 'Perez', '2220-06-05', 'joseph.p@example.com', '9876543305
109 (97, 'Susan', 'Roberts', '2223-03-22', 'susan.r@example.com', '9876543306
110 (98, 'Robert', 'Garcia', '2226-12-09', 'robert.g@example.com', '9876543307
111 (99, 'Katherine', 'Martinez', '2229-09-26', 'katherine.m@example.com', '9876543308
112 (100, 'Steven', 'Hernandez', '2232-06-13', 'steven.h@example.com', '9876543309
113 (101, 'Amanda', 'Lopez', '2235-03-01', 'amanda.l@example.com', '9876543310
114 (102, 'Matthew', 'Wilson', '2238-10-18', 'matthew.w@example.com', '9876543311
115 (103, 'Deborah', 'Anderson', '2241-07-05', 'deborah.a@example.com', '9876543312
116 (104, 'Anthony', 'Taylor', '2244-04-22', 'anthony.t@example.com', '9876543313
117 (105, 'Michelle', 'Thomas', '2247-01-09', 'michelle.t@example.com', '9876543314
118 (106, 'Christopher', 'Jackson', '2250-08-26', 'christopher.j@example.com', '9876543315
119 (107, 'Dorothy', 'White', '2253-05-13', 'dorothy.w@example.com', '9876543316
120 (108, 'Kenneth', 'Harris', '2256-02-01', 'kenneth.h@example.com', '9876543317
121 (109, 'Shirley', 'Clark', '2259-09-18', 'shirley.c@example.com', '9876543318
122 (110, 'Roy', 'Lewis', '2262-06-05', 'roy.l@example.com', '9876543319
123 (111, 'Linda', 'Walker', '2265-03-22', 'linda.w@example.com', '9876543320
124 (112, 'William', 'Hall', '2268-12-09', 'william.h@example.com', '9876543321
125 (113, 'Elizabeth', 'Young', '2271-09-26', 'elizabeth.y@example.com', '9876543322
126 (114, 'Christopher', 'King', '2274-06-13', 'christopher.k@example.com', '9876543323
127 (115, 'Margaret', 'Wright', '2277-03-01', 'margaret.w@example.com', '9876543324
128 (116, 'Daniel', 'Green', '2280-10-18', 'daniel.g@example.com', '9876543325
129 (117, 'Patricia', 'Nelson', '2283-07-05', 'patricia.n@example.com', '9876543326
130 (118, 'Joseph', 'Carter', '2286-04-22', 'joseph.c@example.com', '9876543327
131 (119, 'Susan', 'Mitchell', '2289-01-09', 'susan.m@example.com', '9876543328
132 (120, 'Robert', 'Perez', '2292-08-26', 'robert.p@example.com', '9876543329
133 (121, 'Katherine', 'Roberts', '2295-05-13', 'katherine.r@example.com', '9876543330
134 (122, 'Steven', 'Garcia', '2298-02-01', 'steven.g@example.com', '9876543331
135 (123, 'Amanda', 'Martinez', '2301-09-18', 'amanda.m@example.com', '9876543332
136 (124, 'Matthew', 'Hernandez', '2304-06-05', 'matthew.h@example.com', '9876543333
137 (125, 'Deborah', 'Lopez', '2307-03-22', 'deborah.l@example.com', '9876543334
138 (126, 'Anthony', 'Wilson', '2310-12-09', 'anthony.w@example.com', '9876543335
139 (127, 'Michelle', 'Anderson', '2313-09-26', 'michelle.a@example.com', '9876543336
140 (128, 'Christopher', 'Taylor', '2316-06-13', 'christopher.t@example.com', '9876543337
141 (129, 'Dorothy', 'Thomas', '2319-03-01', 'dorothy.t@example.com', '9876543338
142 (130, 'Kenneth', 'Jackson', '2322-10-18', 'kenneth.j@example.com', '9876543339
143 (131, 'Shirley', 'White', '2325-07-05', 'shirley.w@example.com', '9876543340
144 (132, 'Roy', 'Harris', '2328-04-22', 'roy.h@example.com', '9876543341
145 (133, 'Linda', 'Clark', '2331-01-09', 'linda.c@example.com', '9876543342
146 (134, 'William', 'Lewis', '2334-08-26', 'william.l@example.com', '9876543343
147 (135, 'Elizabeth', 'Walker', '2337-05-13', 'elizabeth.w@example.com', '9876543344
148 (136, 'Christopher', 'Hall', '2340-02-01', 'christopher.h@example.com', '9876543345
149 (137, 'Margaret', 'Young', '2343-09-18', 'margaret.y@example.com', '9876543346
150 (138, 'Daniel', 'King', '2346-06-05', 'daniel.k@example.com', '9876543347
151 (139, 'Patricia', 'Wright', '2349-03-22', 'patricia.w@example.com', '9876543348
152 (140, 'Joseph', 'Green', '2352-12-09', 'joseph.g@example.com', '9876543349
153 (141, 'Susan', 'Nelson', '2355-09-26', 'susan.n@example.com', '9876543350
154 (142, 'Robert', 'Carter', '2358-06-13', 'robert.c@example.com', '9876543351
155 (143, 'Katherine', 'Mitchell', '2361-03-01', 'katherine.m@example.com', '9876543352
156 (144, 'Steven', 'Perez', '2364-10-18', 'steven.p@example.com', '9876543353
157 (145, 'Amanda', 'Roberts', '2367-07-05', 'amanda.r@example.com', '9876543354
158 (146, 'Matthew', 'Garcia', '2370-04-22', 'matthew.g@example.com', '9876543355
159 (147, 'Deborah', 'Martinez', '2373-01-09', 'deborah.m@example.com', '9876543356
160 (148, 'Anthony', 'Hernandez', '2376-08-26', 'anthony.h@example.com', '9876543357
161 (149, 'Michelle', 'Lopez', '2379-05-13', 'michelle.l@example.com', '9876543358
162 (150, 'Christopher', 'Wilson', '2382-02-01', 'christopher.w@example.com', '9876543359
163 (151, 'Dorothy', 'Anderson', '2385-09-18', 'dorothy.a@example.com', '9876543360
164 (152, 'Kenneth', 'Taylor', '2388-06-05', 'kenneth.t@example.com', '9876543361
165 (153, 'Shirley', 'Thomas', '2391-03-22', 'shirley.t@example.com', '9876543362
166 (154, 'Roy', 'Jackson', '2394-12-09', 'roy.j@example.com', '9876543363
167 (155, 'Linda', 'White', '2397-09-26', 'linda.w@example.com', '9876543364
168 (156, 'William', 'Harris', '2400-06-13', 'william.h@example.com', '9876543365
169 (157, 'Elizabeth', 'Clark', '2403-03-01', 'elizabeth.c@example.com', '9876543366
170 (158, 'Christopher', 'Lewis', '2406-10-18', 'christopher.l@example.com', '9876543367
171 (159, 'Margaret', 'Walker', '2409-07-05', 'margaret.w@example.com', '9876543368
172 (160, 'Daniel', 'Hall', '2412-04-22', 'daniel.h@example.com', '9876543369
173 (161, 'Patricia', 'Young', '2415-01-09', 'patricia.y@example.com', '9876543370
174 (162, 'Joseph', 'King', '2418-08-26', 'joseph.k@example.com', '9876543371
175 (163, 'Susan', 'Wright', '2421-05-13', 'susan.w@example.com', '9876543372
176 (164, 'Robert', 'Green', '2424-02-01', 'robert.g@example.com', '9876543373
177 (165, 'Katherine', 'Nelson', '2427-09-18', 'katherine.n@example.com', '9876543374
178 (166, 'Steven', 'Carter', '2430-06-05', 'steven.c@example.com', '9876543375
179 (167, 'Amanda', 'Mitchell', '2433-03-22', 'amanda.m@example.com', '9876543376
180 (168, 'Matthew', 'Perez', '2436-12-09', 'matthew.p@example.com', '9876543377
181 (169, 'Deborah', 'Roberts', '2439-09-26', 'deborah.r@example.com', '9876543378
182 (170, 'Anthony', 'Garcia', '2442-06-13', 'anthony.g@example.com', '9876543379
183 (171, 'Michelle', 'Martinez', '2445-03-01', 'michelle.m@example.com', '9876543380
184 (172, 'Christopher', 'Hernandez', '2448-10-18', 'christopher.h@example.com', '9876543381
185 (173, 'Dorothy', 'Lopez', '2451-07-05', 'dorothy.l@example.com', '9876543382
186 (174, 'Kenneth', 'Wilson', '2454-04-22', 'kenneth.w@example.com', '9876543383
187 (175, 'Shirley', 'Anderson', '2457-01-09', 'shirley.a@example.com', '9876543384
188 (176, 'Roy', 'Taylor', '2460-08-26', 'roy.t@example.com', '9876543385
189 (177, 'Linda', 'Thomas', '2463-05-13', 'linda.t@example.com', '9876543386
190 (178, 'William', 'Jackson', '2466-02-01', 'william.j@example.com', '9876543387
191 (179, 'Elizabeth', 'White', '2469-09-18', 'elizabeth.w@example.com', '9876543388
192 (180, 'Christopher', 'Harris', '2472-06-05', 'christopher.h@example.com', '9876543389
193 (181, 'Margaret', 'Clark', '2475-03-22', 'margaret.c@example.com', '9876543390
194 (182, 'Daniel', 'Lewis', '2478-12-09', 'daniel.l@example.com', '9876543391
195 (183, 'Patricia', 'Walker', '2481-09-26', 'patricia.w@example.com', '9876543392
196 (184, 'Joseph', 'Hall', '2484-06-13', 'joseph.h@example.com', '9876543393
197 (185, 'Susan', 'Young', '2487-03-01', 'susan.y@example.com', '9876543394
198 (186, 'Robert', 'King', '2490-10-18', 'robert.k@example.com', '9876543395
199 (187, 'Katherine', 'Wright', '2493-07-05', 'katherine.w@example.com', '9876543396
200 (188, 'Steven', 'Green', '2496-04-22', 'steven.g@example.com', '9876543397
201 (189, 'Amanda', 'Nelson', '2499-01-09', 'amanda.n@example.com', '9876543398
202 (190, 'Matthew', 'Carter', '2502-08-26', 'matthew.c@example.com', '9876543399
203 (191, 'Deborah', 'Mitchell', '2505-05-13', 'deborah.m@example.com', '9876543400
204 (192, 'Anthony', 'Perez', '2508-02-01', 'anthony.p@example.com', '9876543401
205 (193, 'Michelle', 'Roberts', '2511-09-18', 'michelle.r@example.com', '9876543402
206 (194, 'Christopher', 'Garcia', '2514-06-05', 'christopher.g@example.com', '9876543403
207 (195, 'Dorothy', 'Martinez', '2517-03-22', 'dorothy.m@example.com', '9876543404
208 (196, 'Kenneth', 'Hernandez', '2520-12-09', 'kenneth.h@example.com', '9876543405
209 (197, 'Shirley', 'Lopez', '2523-09-26', 'shirley.l@example.com', '9876543406
210 (198, 'Roy', 'Wilson', '2526-06-13', 'roy.w@example.com', '9876543407
211 (199, 'Linda', 'Anderson', '2529-03-01', 'linda.a@example.com', '9876543408
212 (200, 'William', 'Taylor', '2532-10-18', 'william.t@example.com', '9876543409
213 (201, 'Elizabeth', 'Thomas', '2535-07-05', 'elizabeth.t@example.com', '9876543410
214 (202, 'Christopher', 'Jackson', '2538-04-22', 'christopher.j@example.com', '9876543411
215 (203, 'Margaret', 'White', '2541-01-09', 'margaret.w@example.com', '9876543412
216 (204, 'Daniel', 'Harris', '2544-08-26', 'daniel.h@example.com', '9876543413
217 (205, 'Patricia', 'Clark', '2547-05-13', 'patricia.c@example.com', '9876543414
218 (206, 'Joseph', 'Lewis', '2550-02-01', 'joseph.l@example.com', '9876543415
219 (207, 'Susan', 'Walker', '2553-09-18', 'susan.w@example.com', '9876543416
220 (208, 'Robert', 'Hall', '2556-06-05', 'robert.h@example.com', '9876543417
221 (209, 'Katherine', 'Young', '2559-03-22', 'katherine.y@example.com', '9876543418
222 (210, 'Steven', 'King', '2562-12-09', 'steven.k@example.com', '9876543419
223 (211, 'Amanda', 'Wright', '2565-09-26', 'amanda.w@example.com', '9876543420
224 (212, 'Matthew', 'Green', '2568-06-13', 'matthew.g@example.com', '9876543421
225 (213, 'Deborah', 'Nelson', '2571-03-01', 'deborah.n@example.com', '9876543422
226 (214, 'Anthony', 'Carter', '2574-10-18', 'anthony.c@example.com', '9876543423
227 (215, 'Michelle', 'Mitchell', '2577-07-05', 'michelle.m@example.com', '9876543424
228 (216, 'Christopher', 'Perez', '2580-04-22', 'christopher.p@example.com', '9876543425
229 (217, 'Dorothy', 'Roberts', '2583-01-09', 'dorothy.r@example.com', '9876543426
230 (218, 'Kenneth', 'Garcia', '2586-08-26', 'kenneth.g@example.com', '9876543427
231 (219, 'Shirley', 'Martinez', '2589-05-13', 'shirley.m@example.com', '9876543428
232 (220, 'Roy', 'Hernandez', '2592-02-01', 'roy.h@example.com', '9876543429
233 (221, 'Linda', 'Lopez', '2595-09-18', 'linda.l@example.com', '9876543430
234 (222, 'William', 'Wilson', '2598-06-05', 'william.w@example.com', '9876543431
235 (223, 'Elizabeth', 'Anderson', '2601-03-22', 'elizabeth.a@example.com', '9876543432
236 (224, 'Christopher', 'Taylor', '2604-12-09', 'christopher.t@example.com', '9876543433
237 (225, 'Margaret', 'Thomas', '2607-09-26', 'margaret.t@example.com', '9876543434
238 (226, 'Daniel', 'Jackson', '2610-06-13', 'daniel.j@example.com', '9876543435
239 (227, 'Patricia', 'White', '2613-03-01', 'patricia.w@example.com', '9876543436
240 (228, 'Joseph', 'Harris', '2616-10-18', 'joseph.h@example.com', '9876543437
241 (229, 'Susan', 'Clark', '2619-07-05', 'susan.c@example.com', '9876543438
242 (230, 'Robert', 'Lewis', '2622-04-22', 'robert.l@example.com', '9876543439
243 (231, 'Katherine', 'Walker', '2625-01-09', 'katherine.w@example.com', '9876543440
244 (232, 'Steven', 'Hall', '2628-08-26', 'steven.h@example.com', '9876543441
245 (233, 'Amanda', 'Young', '2631-05-13', 'amanda.y@example.com', '9876543442
246 (234, 'Matthew', 'King', '2634-02-01', 'matthew.k@example.com', '9876543443
247 (235, 'Deborah', 'Wright', '2637-09-18', 'deborah.w@example.com', '9876543444
248 (236, 'Anthony', 'Green', '2640-06-05', 'anthony.g@example.com', '9876543445
249 (237, 'Michelle', 'Nelson', '2643-03-22', 'michelle.n@example.com', '9876543446
250 (238, 'Christopher', 'Carter', '2646-12-09', 'christopher.c@example.com', '9876543447
251 (239, 'Dorothy', 'Mitchell', '2649-09-26', 'dorothy.m@example.com', '9876543448
252 (240, 'Kenneth', 'Perez', '2652-06-13', 'kenneth.p@example.com', '9876543449
253 (241, 'Shirley', 'Roberts', '2655-03-01', 'shirley.r@example.com', '9876543450
254 (242, 'Roy', 'Garcia', '2658-10-18', 'roy.g@example.com', '9876543451
255 (243, 'Linda', 'Martinez', '2661-07-05', 'linda.m@example.com', '9876543452
256 (244, 'William', 'Hernandez', '2664-04-22', 'william.h@example.com', '9876543453
257 (245, 'Elizabeth', 'Lopez', '2667-01-09', 'elizabeth.l@example.com', '9876543454
258 (246, 'Christopher', 'Wilson', '2670-08-26', 'christopher.w@example.com', '9876543455
259 (247, 'Margaret', 'Anderson', '2673-05-13', 'margaret.a@example.com', '9876543456
260 (248, 'Daniel', 'Taylor', '2676-02-01', 'daniel.t@example.com', '9876543457
261 (249, 'Patricia', 'Thomas', '2679-09-18', 'patricia.t@example.com', '9876543458
262 (250, 'Joseph', 'Jackson', '2682-06-05', 'joseph.j@example.com', '9876543459
263 (251, 'Susan', 'White', '2685-03-22', 'susan.w@example.com', '9876543460
264 (252, 'Robert', 'Harris', '2688-12-09', 'robert.h@example.com', '9876543461
265 (253, 'Katherine', 'Clark', '2691-09-26', 'katherine.c@example.com', '9876543462
266 (254, 'Steven', 'Lewis', '2694-06-13', 'steven.l@example.com', '9876543463
267 (255, 'Amanda', 'Walker', '2697-03-01', 'amanda.w@example.com', '9876543464
268 (256, 'Matthew', 'Hall', '2700-10-18', 'matthew.h@example.com', '9876543465
269 (257, 'Deborah', 'Young', '2703-07-05', 'deborah.y@example.com', '9876543466
270 (258, 'Anthony', 'King', '2706-04-22', 'anthony.k@example.com', '9876543467
271 (259, 'Michelle', 'Wright', '2709-01-09', 'michelle.w@example.com', '9876543468
272 (260, 'Christopher', 'Green', '2712-08-26', 'christopher.g@example.com', '9876543469
273 (261, 'Dorothy', 'Nelson', '2715-05-13', 'dorothy.n@example.com', '9876543470
274 (262, 'Kenneth', 'Carter', '2718-02-01', 'kenneth.c@example.com', '9876543471
275 (263, 'Shirley', 'Mitchell', '2721-09-18', 'shirley.m@example.com', '9876543472
276 (264, 'Roy', 'Perez', '2724-06-05', 'roy.p@example.com', '9876543473
277 (265, 'Linda', 'Roberts', '2727-03-22', 'linda.r@example.com', '9876543474
278 (266, 'William', 'Garcia', '2730-12-09', 'william.g@example.com', '9876543475
279 (267, 'Elizabeth', 'Martinez', '2733-09-26', 'elizabeth.m@example.com', '9876543476
280 (268, 'Christopher', 'Hernandez', '2736-06-13', 'christopher.h@example.com', '9876543477
281 (269, 'Margaret', 'Lopez', '2739-03-01', 'margaret.l@example.com', '9876543478
282 (270, 'Daniel', 'Wilson', '2742-10-18', 'daniel.w@example.com', '9876543479
283 (271, 'Patricia', 'Anderson', '2745-07-05', 'patricia.a@example.com', '98765434
```

```

52 • INSERT INTO Transactions VALUES
53     (1001, 101, 'deposit', 5000.00, '2025-04-01 10:00:00'),
54     (1002, 102, 'withdrawal', 2000.00, '2025-04-02 11:30:00'),
55     (1003, 103, 'deposit', 500.00, '2025-04-03 09:15:00'),
56     (1004, 104, 'transfer', 1000.00, '2025-04-04 14:00:00'),
57     (1005, 105, 'deposit', 2500.00, '2025-04-05 12:45:00'),
58     (1006, 106, 'withdrawal', 1500.00, '2025-04-06 16:20:00'),
59     (1007, 107, 'deposit', 100.00, '2025-04-07 08:10:00'),
60     (1008, 108, 'transfer', 2000.00, '2025-04-07 18:25:00'),
61     (1009, 109, 'deposit', 3000.00, '2025-04-08 10:50:00'),
62     (1010, 110, 'withdrawal', 500.00, '2025-04-08 13:15:00');

```

Result Grid					
Filter Rows:					
Edit:					
Export/Import:					
	transaction_id	account_id	transaction_type	amount	transaction_date
▶	1001	101	deposit	5000.00	2025-04-01 10:00:00
	1002	102	withdrawal	2000.00	2025-04-02 11:30:00
	1003	103	deposit	500.00	2025-04-03 09:15:00
	1004	104	transfer	1000.00	2025-04-04 14:00:00
	1005	105	deposit	2500.00	2025-04-05 12:45:00
	1006	106	withdrawal	1500.00	2025-04-06 16:20:00
	1007	107	deposit	100.00	2025-04-07 08:10:00
	1008	108	transfer	2000.00	2025-04-07 18:25:00
	1009	109	deposit	3000.00	2025-04-08 10:50:00
	1010	110	withdrawal	500.00	2025-04-08 13:15:00
▲	NULL	NULL	NULL	NULL	NULL

2. Write SQL queries for the following tasks:


1. Write a SQL query to retrieve the name, account type and email of all customers.

```
select *from Transactions;
```

```

SELECT
    c.first_name,
    c.last_name,
    a.account_type,
    c.email
FROM
    Customers c
JOIN
    Accounts a ON c.customer_id = a.customer_id;

```


Result Grid				
Filter Rows:		Export:  Wrap Cell Cor		
	first_name	last_name	account_type	email
▶	John	Doe	savings	john.doe@example.com
	Jane	Smith	current	jane.smith@example.com
	Alice	Johnson	zero_balance	alice.j@example.com
	Bob	Brown	savings	bob.b@example.com
	Charlie	Davis	current	charlie.d@example.com
	Emily	Clark	savings	emily.c@example.com
	Frank	Wright	zero_balance	frank.w@example.com
	Grace	Lee	savings	grace.l@example.com
	Hank	Green	current	hank.g@example.com
	Ivy	King	savings	ivy.k@example.com

2. Write a SQL query to list all transaction corresponding customer.

```

SELECT
    c.first_name,
    c.last_name,
    t.transaction_id,
    t.transaction_type,
    t.amount,
    t.transaction_date
FROM
    Customers c
JOIN
    Accounts a ON c.customer_id = a.customer_id
JOIN
    Transactions t ON a.account_id = t.account_id;

```

	first_name	last_name	transaction_id	transaction_type	amount	transaction_date
▶	John	Doe	1001	deposit	5000.00	2025-04-01 10:00:00
	Jane	Smith	1002	withdrawal	2000.00	2025-04-02 11:30:00
	Alice	Johnson	1003	deposit	500.00	2025-04-03 09:15:00
	Bob	Brown	1004	transfer	1000.00	2025-04-04 14:00:00
	Charlie	Davis	1005	deposit	2500.00	2025-04-05 12:45:00
	Emily	Clark	1006	withdrawal	1500.00	2025-04-06 16:20:00
	Frank	Wright	1007	deposit	100.00	2025-04-07 08:10:00
	Grace	Lee	1008	transfer	2000.00	2025-04-07 18:25:00
	Hank	Green	1009	deposit	3000.00	2025-04-08 10:50:00
	Ivy	King	1010	withdrawal	500.00	2025-04-08 13:15:00

3. Write a SQL query to increase the balance of a specific account by a certain amount.

```
90 • UPDATE Accounts
91     SET balance = balance + 500
92     WHERE account_id = 101;
93 • select balance from accounts;
94
```

Result Grid | Filter Rows: | Export: | Wrap

balance
16500.00
25000.50
500.00
10000.00
20000.75
30000.00
0.00
12000.00
8000.80
5000.00

4. Write a SQL query to Combine first and last names of customers as a full_name.

```
95 • SELECT
96     CONCAT(first_name, ' ', last_name) AS full_name
97 FROM
98     Customers;
99
100
```

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

full_name
John Doe
Jane Smith
Alice Johnson
Bob Brown
Charlie Davis
Emily Clark
Frank Wright
Grace Lee
Hank Green
Ivy King

5. Write a SQL query to remove accounts with a balance of zero where the account type is savings.

```
DELETE FROM Accounts
WHERE balance = 0 AND account_type = 'savings';
```

6. Write a SQL query to Find customers living in a specific city.

```
105 • SELECT * FROM Customers
106 WHERE address LIKE '%Chennai%';
```

Result Grid

Filter Rows:

Edit:

Export/Import:

	customer_id	first_name	last_name	DOB	email	phone_number	address
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

7. Write a SQL query to Get the account balance for a specific account.

```
108 • SELECT balance
109 FROM Accounts
110 WHERE account_id = 101;
111
```

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

balance
16500.00

8. Write a SQL query to List all current accounts with a balance greater than \$1,000.

```
112 • SELECT * FROM Accounts
113 WHERE account_type = 'current' AND balance > 1000;
114
```

Result Grid | Filter Rows: | Edit: | Export/Import:

	account_id	customer_id	account_type	balance
▶	102	2	current	25000.50
	105	5	current	20000.75
	109	9	current	8000.80
*	NULL	NULL	NULL	NULL

9. Write a SQL query to Retrieve all transactions for a specific account.



```
115 • SELECT * FROM Transactions
116 WHERE account_id = 101
117 ;
118
```

Result Grid | Filter Rows: | Edit: | Export/Import:

	transaction_id	account_id	transaction_type	amount	transaction_date
▶	1001	101	deposit	5000.00	2025-04-01 10:00:00
*	NULL	NULL	NULL	NULL	NULL




10. Write a SQL query to Calculate the interest accrued on savings accounts based on a given interest rate.

```
120 • SELECT account_id, balance, balance * 0.04 AS interest
121 FROM Accounts
122 WHERE account_type = 'savings';
123
```

Result Grid			
Filter Rows: <input type="text"/>			
Export:  Wrap Cell Content: 			
	account_id	balance	interest
▶	101	16500.00	660.0000
	104	10000.00	400.0000
	106	30000.00	1200.0000
	108	12000.00	480.0000
	110	5000.00	200.0000

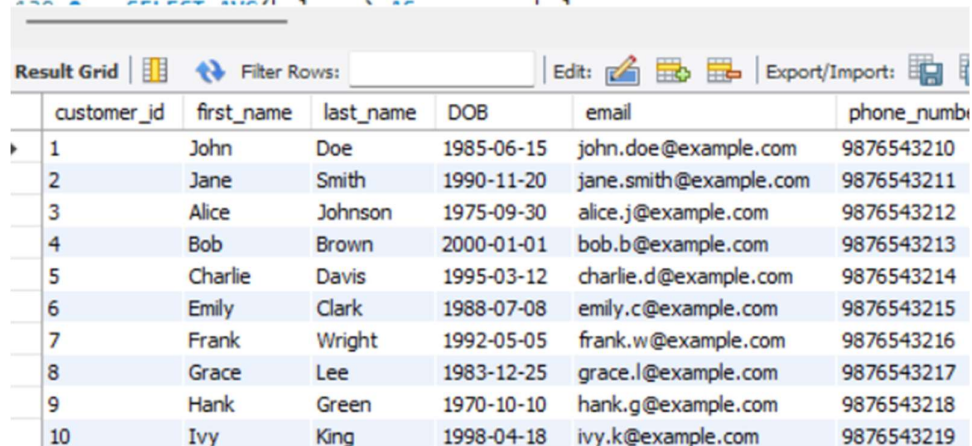
11. Write a SQL query to Identify accounts where the balance is less than a specified overdraft limit.

```
124 • SELECT *FROM Accounts
125 WHERE balance < 500;
126
```

Result Grid				
Filter Rows: <input type="text"/>				
Edit:   				
	account_id	customer_id	account_type	balance
▶	107	7	zero_balance	0.00
*	NULL	NULL	NULL	NULL

12. Write a SQL query to Find customers not living in a specific city.

```
127 • SELECT *FROM Customers
128 WHERE address NOT LIKE '%Chennai%';
129
```



The screenshot shows a SQL query result grid with 10 rows of customer data. The columns are customer_id, first_name, last_name, DOB, email, and phone_number. The data is as follows:

customer_id	first_name	last_name	DOB	email	phone_number
1	John	Doe	1985-06-15	john.doe@example.com	9876543210
2	Jane	Smith	1990-11-20	jane.smith@example.com	9876543211
3	Alice	Johnson	1975-09-30	alice.j@example.com	9876543212
4	Bob	Brown	2000-01-01	bob.b@example.com	9876543213
5	Charlie	Davis	1995-03-12	charlie.d@example.com	9876543214
6	Emily	Clark	1988-07-08	emily.c@example.com	9876543215
7	Frank	Wright	1992-05-05	frank.w@example.com	9876543216
8	Grace	Lee	1983-12-25	grace.l@example.com	9876543217
9	Hank	Green	1970-10-10	hank.g@example.com	9876543218
10	Ivy	King	1998-04-18	ivy.k@example.com	9876543219

Tasks 3: Aggregate functions, Having, Order By, Group By and Joins:

1. Write a SQL query to Find the average account balance for all customers.

```
130 • SELECT AVG(balance) AS average_balance
131 FROM Accounts;
132
```



The screenshot shows a SQL query result grid with one row representing the average account balance. The column is average_balance and the value is 12700.205000.

average_balance
12700.205000

2. Write a SQL query to Retrieve the top 10 highest account balances.

```
133 • SELECT *FROM Accounts
134 ORDER BY balance DESC
135 LIMIT 10;
136
```

Result Grid

	account_id	customer_id	account_type	balance
▶	106	6	savings	30000.00
	102	2	current	25000.50
	105	5	current	20000.75
	101	1	savings	16500.00
	108	8	savings	12000.00
	104	4	savings	10000.00
	109	9	current	8000.80
	110	10	savings	5000.00
	103	3	zero_balance	500.00
	107	7	zero_balance	0.00
	NULL	NULL	NULL	NULL

3. Write a SQL query to Calculate Total Deposits for All Customers in specific date.

```
137 • SELECT SUM(amount) AS total_deposits
138 FROM Transactions
139 WHERE transaction_type = 'deposit'
140 AND DATE(transaction_date) = '2025-04-07';
141
```

Result Grid

	account_id	customer_id	account_type	balance
▶	106	6	savings	30000.00
	102	2	current	25000.50
	105	5	current	20000.75
	101	1	savings	16500.00
	108	8	savings	12000.00
	104	4	savings	10000.00
	109	9	current	8000.80
	110	10	savings	5000.00
	103	3	zero_balance	500.00
	107	7	zero_balance	0.00
	NULL	NULL	NULL	NULL

4. Write a SQL query to Find the Oldest and Newest Customers.

```
142 • SELECT * FROM Customers
143 ORDER BY DOB ASC
144 LIMIT 1;
145 • SELECT * FROM Customers
146 ORDER BY DOB DESC
147 LIMIT 1;
```

customer_id	first_name	last_name	DOB	email	phone_number	add
4	Bob	Brown	2000-01-01	bob.b@example.com	9876543213	101
NULL	NULL	NULL	NULL	NULL	NULL	NULL

5. Write a SQL query to Retrieve transaction details along with the account type.

```
149 • SELECT t.*, a.account_type
150 FROM Transactions t
151 JOIN Accounts a ON t.account_id = a.account_id;
```

transaction_id	account_id	transaction_type	amount	transaction_date	account_type
1001	101	deposit	5000.00	2025-04-01 10:00:00	savings
1002	102	withdrawal	2000.00	2025-04-02 11:30:00	current
1003	103	deposit	500.00	2025-04-03 09:15:00	zero_balance
1004	104	transfer	1000.00	2025-04-04 14:00:00	savings
1005	105	deposit	2500.00	2025-04-05 12:45:00	current
1006	106	withdrawal	1500.00	2025-04-06 16:20:00	savings
1007	107	deposit	100.00	2025-04-07 08:10:00	zero_balance
1008	108	transfer	2000.00	2025-04-07 18:25:00	savings
1009	109	deposit	3000.00	2025-04-08 10:50:00	current
1010	110	withdrawal	500.00	2025-04-08 13:15:00	savings

6. Write a SQL query to Get a list of customers along with their account details.

```
155 • SELECT c.*, a.account_id, a.account_type, a.balance
156 FROM Customers c
157 JOIN Accounts a ON c.customer_id = a.customer_id;
```

Result Grid						
Filter Rows: <input type="text"/> Export: Wrap Cell Content:						
	customer_id	first_name	last_name	DOB	email	phone_number
▶	1	John	Doe	1985-06-15	john.doe@example.com	9876543210
	2	Jane	Smith	1990-11-20	jane.smith@example.com	9876543211
	3	Alice	Johnson	1975-09-30	alice.j@example.com	9876543212
	4	Bob	Brown	2000-01-01	bob.b@example.com	9876543213
	5	Charlie	Davis	1995-03-12	charlie.d@example.com	9876543214
	6	Emily	Clark	1988-07-08	emily.c@example.com	9876543215
	7	Frank	Wright	1992-05-05	frank.w@example.com	9876543216
	8	Grace	Lee	1983-12-25	grace.l@example.com	9876543217
	9	Hank	Green	1970-10-10	hank.g@example.com	9876543218
	10	Ivy	King	1998-04-18	ivy.k@example.com	9876543219

7. Write a SQL query to Retrieve transaction details along with customer information for a specific account.

```
159 • SELECT t.*, c.first_name, c.last_name, c.email
160 FROM Transactions t
161 JOIN Accounts a ON t.account_id = a.account_id
162 JOIN Customers c ON a.customer_id = c.customer_id
163 WHERE t.account_id = 101;
164
```

Result Grid						
Filter Rows: <input type="text"/> Export: Wrap Cell Content:						
	transaction_id	account_id	transaction_type	amount	transaction_date	first_name
▶	1001	101	deposit	5000.00	2025-04-01 10:00:00	John

8. Write a SQL query to Identify customers who have more than one account.

```
165 • SELECT customer_id, COUNT(*) AS account_count
166 FROM Accounts
167 GROUP BY customer_id
168 HAVING COUNT(*) > 1;
169
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
customer_id	account_count		

9. Write a SQL query to Calculate the difference in transaction amounts between deposits and withdrawals.

```
170 • SELECT
171 (SUM(CASE WHEN transaction_type = 'deposit' THEN amount ELSE 0 END) -
172 SUM(CASE WHEN transaction_type = 'withdrawal' THEN amount ELSE 0 END))
173 FROM Transactions;
174
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
net_difference			
7100.00			

Result Grid

Form Editor

Field Types

10. Write a SQL query to Calculate the average daily balance for each account over a specified period.

```
176 • SELECT
177     account_id,
178     AVG(amount) AS avg_transaction_amount
179 FROM Transactions
180 WHERE transaction_date BETWEEN '2025-04-01' AND '2025-04-07'
181 GROUP BY account_id;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
account_id	avg_transaction_amount		
101	5000.000000		
102	2000.000000		
103	500.000000		
104	1000.000000		
105	2500.000000		
106	1500.000000		

11. Calculate the total balance for each account type.

```
183 • SELECT account_type, SUM(balance) AS total_balance
184 FROM Accounts
185 GROUP BY account_type;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
account_type	total_balance		
savings	73500.00		
current	53002.05		
zero_balance	500.00		

12. Identify accounts with the highest number of transactions order by descending order.

```
187 • SELECT account_id, COUNT(*) AS transaction_count
188 FROM Transactions
189 GROUP BY account_id
190 ORDER BY transaction_count DESC;
191
```

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

	account_id	transaction_count
▶	101	1
	102	1
	103	1
	104	1
	105	1
	106	1
	107	1
	108	1
	109	1
	110	1

13. List customers with high aggregate account balances, along with their account types.

```
192 • SELECT c.customer_id, c.first_name, c.last_name, a.account_type, SUM(a.ba
193 FROM Customers c
194 JOIN Accounts a ON c.customer_id = a.customer_id
195 GROUP BY c.customer_id, a.account_type
196 HAVING SUM(a.balance) > 20000;
197
```

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

	customer_id	first_name	last_name	account_type	total_balance
▶	2	Jane	Smith	current	25000.50
	5	Charlie	Davis	current	20000.75
	6	Emily	Clark	savings	30000.00

Result Grid
Form Editor
Field Types

14. Identify and list duplicate transactions based on transaction amount, date, and account.

```
198 • SELECT account_id, amount, DATE(transaction_date) AS trans_date, COUNT(*)
199 FROM Transactions
200 GROUP BY account_id, amount, DATE(transaction_date)
201 HAVING COUNT(*) > 1;
202
```

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

account_id	amount	trans_date	duplicate_count
------------	--------	------------	-----------------

Result Grid
Form Editor
Field Types

Tasks 4: Subquery and its type:

1. Retrieve the customer(s) with the highest account balance.

```
203 • SELECT c.customer_id, c.first_name, c.last_name, a.balance
204 FROM Customers c
205 JOIN Accounts a ON c.customer_id = a.customer_id
206 WHERE a.balance = (SELECT MAX(balance) FROM Accounts);
207
```

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

customer_id	first_name	last_name	balance
6	Emily	Clark	30000.00

2. Calculate the average account balance for customers who have more than one account.

```
208 • SELECT AVG(balance) AS avg_balance
209 FROM Accounts
210 WHERE customer_id IN (
211     SELECT customer_id
212     FROM Accounts
213     GROUP BY customer_id
214     HAVING COUNT(account_id) > 1
215 )
```

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

avg_balance
NULL

3. Retrieve accounts with transactions whose amounts exceed the average transaction amount.

```
217 • SELECT DISTINCT t.account_id, t.transaction_id, t.amount
218 FROM Transactions t
219 WHERE t.amount > (SELECT AVG(amount) FROM Transactions);
220
```

Result Grid | Filter Rows: | Edit: | Export/Import: |

account_id	transaction_id	amount
101	1001	5000.00
102	1002	2000.00
105	1005	2500.00
108	1008	2000.00
109	1009	3000.00
NULL	NULL	NULL

6. Retrieve transactions for accounts with the lowest balance.

```
232 • SELECT t.*
233 FROM Transactions t
234 WHERE t.account_id = (
235     SELECT account_id
236     FROM Accounts
237     ORDER BY balance ASC
238     LIMIT 1
239 )
```

Result Grid

	transaction_id	account_id	transaction_type	amount	transaction_date
▶	1007	107	deposit	100.00	2025-04-07 08:10:00
*	NULL	NULL	NULL	NULL	NULL

7. Identify customers who have accounts of multiple types.

```
241 • SELECT customer_id
242 FROM Accounts
243 GROUP BY customer_id
244 HAVING COUNT(DISTINCT account_type) > 1;
245
```

Result Grid		 Filter Rows: <input type="text"/>	Export: 	Wrap Cell Content: 
customer_id				

8. Calculate the percentage of each account type out of the total number of accounts.

```

246 • SELECT account_type,
247         COUNT(*) AS type_count,
248         ROUND(100.0 * COUNT(*) / (SELECT COUNT(*) FROM Accounts), 2) AS pe
249 FROM Accounts
250 GROUP BY account_type;

```

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

	account_type	type_count	percentage
▶	savings	5	50.00
	current	3	30.00
	zero_balance	2	20.00

Result 38 x Read Only

9. Retrieve all transactions for a customer with a given customer_id.

```

260 • SELECT t.*
261 FROM Transactions t
262 JOIN Accounts a ON t.account_id = a.account_id
263 WHERE a.customer_id=10;
264
265 • SELECT DISTINCT account_type,
266         (SELECT SUM(balance)

```

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

	transaction_id	account_id	transaction_type	amount	transaction_date
▶	1010	110	withdrawal	500.00	2025-04-08 13:15:00


10. Calculate the total balance for each account type, including a subquery within the SELECT clause.

```

265 • SELECT DISTINCT account_type,
266     (SELECT SUM(balance)
267      FROM Accounts a2
268      WHERE a2.account_type = a1.account_type) AS total_balance
269 FROM Accounts a1;

```

Result Grid	
account_type	total_balance
savings	73500.00
current	53002.05
zero_balance	500.00

Export:  Wrap Cell Content: 