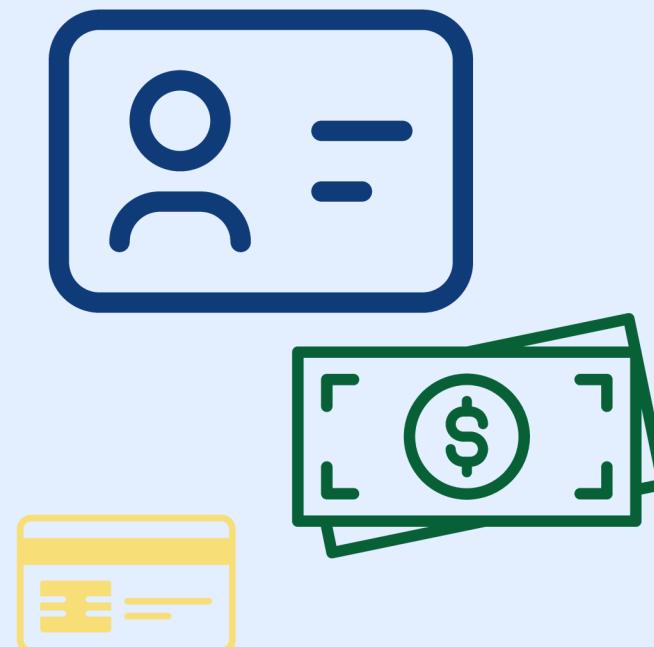


BANK DATABASE



Group members:

موده سعد الوزيناني

444001988

فرح محمد البقمي

443009530

دانيا خالد باسلوم

44411990

Section number:

6

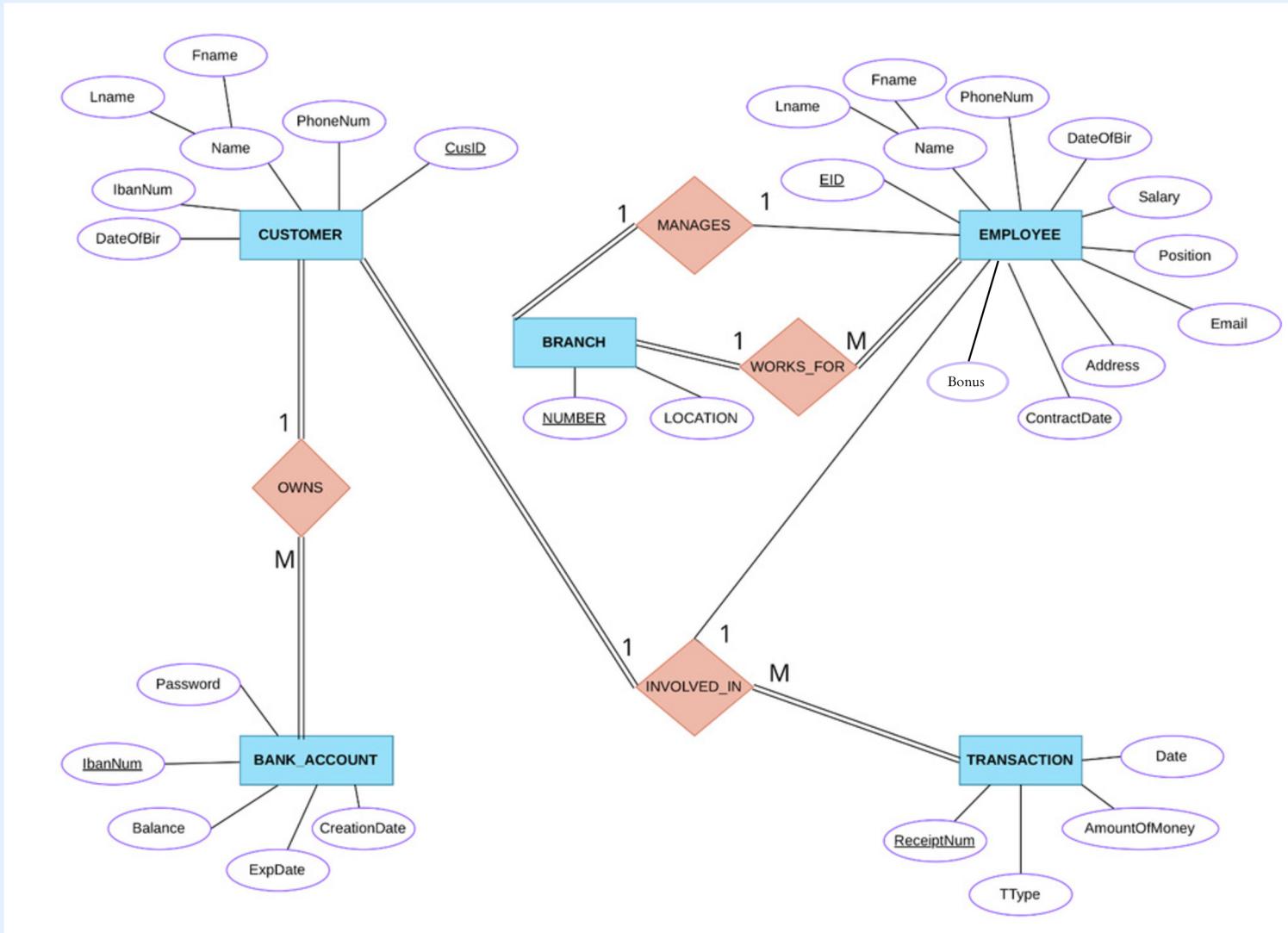
Group work report:

	موده	فرج	دانیا
Business Rules	60%	20%	20%
Chen's Notation	20%	60%	20%
UML Notation	20%	20%	60%
Mapping	60%	20%	20%
Normalization	20%	60%	20%
SQL tables : employee /emp_position /branch	5%	90%	5%
SQL tables : bankAccount /Transaction	90%	5%	5%
SQL tables : Customer /InvolvedIn	5%	5%	90%
Discussion	37%	37%	26%

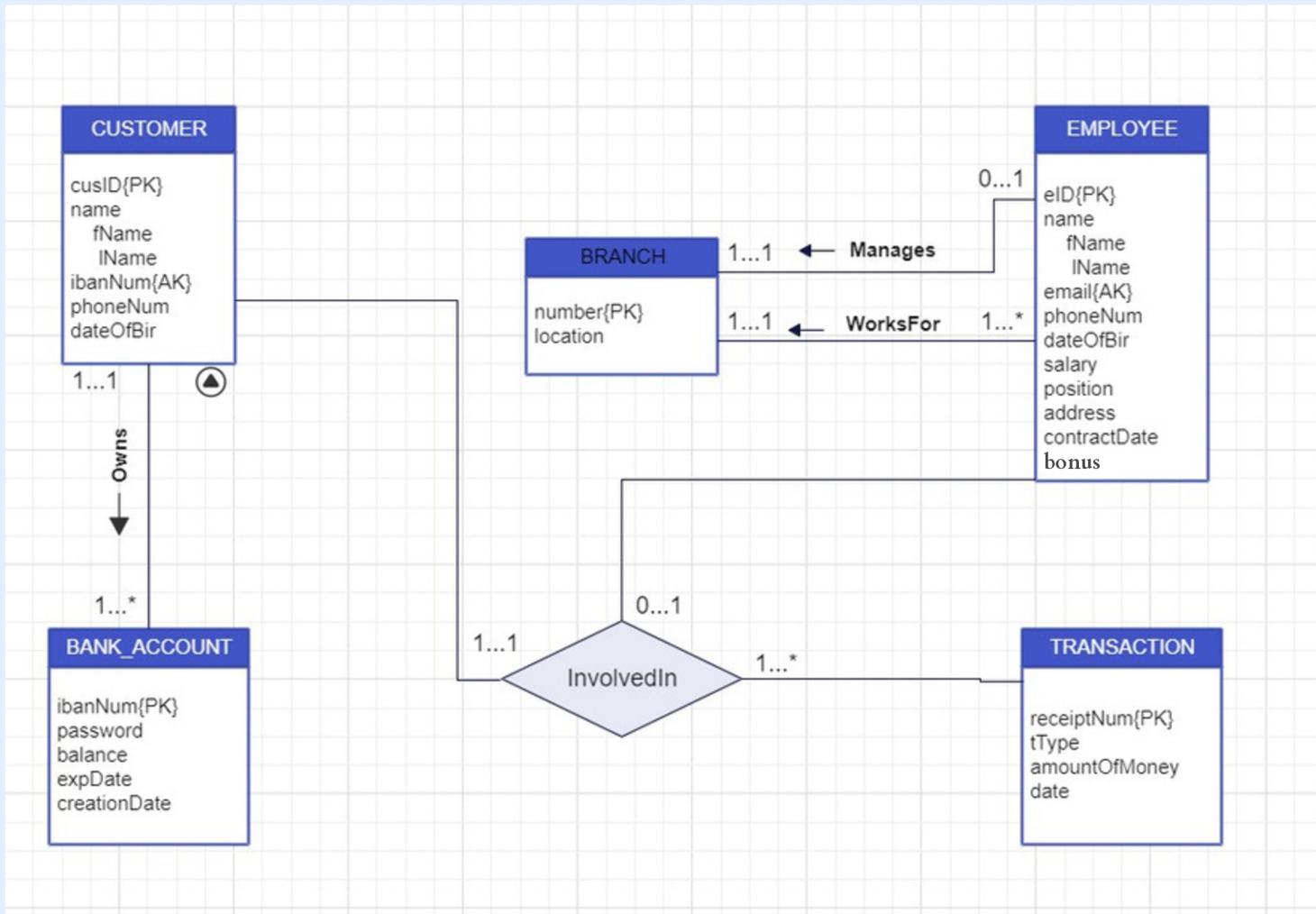
Business Rules:

- The bank has many BRANCHes, and each BRANCH has a number (unique value),location, and an employee who manages the branch.
- The data base will store each EMPLOYEE's name and ID (unique value), address, salary, phone number, date of birth, contract date, email, position, bonus. There are many employees and each employee works for one branch but it's not necessary for employees to work on the transactions.
- The bank has multiple CUSTOMERs and the data base will store each CUSTOMERs name, ID, phone number, IBAN number, date of birth. Each customer can have multiple bank accounts and do many transactions.
- All BANK_ACCOUNTs are going to be stored in the data base including its IBAN number (unique value), pass word, balance, expire date, creation date. And each bank account should be owned by a customer.
- The TRANSACTIONs will include the receipt number (unique value), transaction type, amount of money, date. All transactions should be done by the customer and there could be an employee to help with it.

Chen's Notation:



UML Notation:



Mapping:

Step 1:
Mapping of regular entity types

Customer

<u>CusID</u>	DateofBir	IbanNum	Fname	Lname	PhonNum
--------------	-----------	---------	-------	-------	---------

BankAccount

<u>IbanNum</u>	Password	Balance	ExpDate	CreationDate
----------------	----------	---------	---------	--------------

Branch

<u>Number</u>	Location
---------------	----------

Employee

<u>EID</u>	Ename	Lname	PhoneNum	DateofBir	Salary	Position	Email	Address	ContextDate	Bonus
------------	-------	-------	----------	-----------	--------	----------	-------	---------	-------------	-------

Transaction

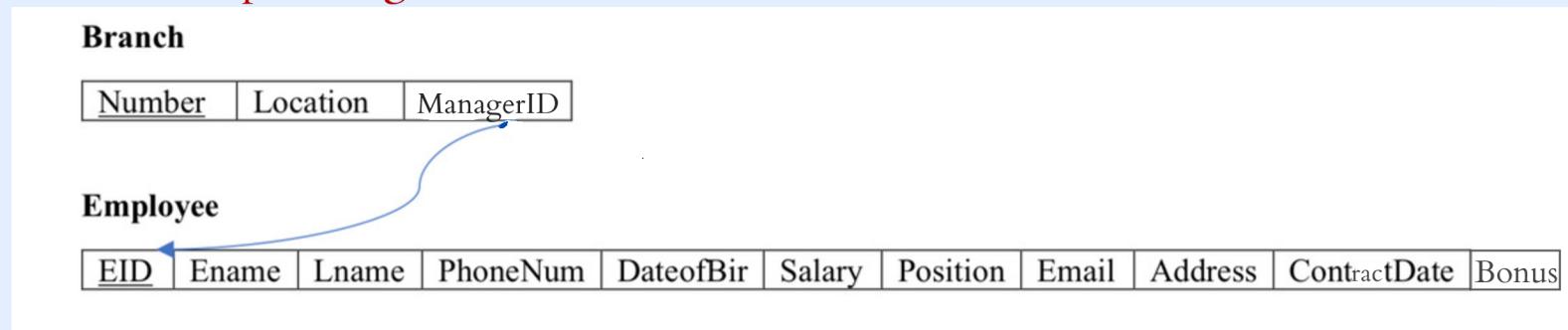
<u>ReceiptNum</u>	Ttype	AmountOfMoney	Date
-------------------	-------	---------------	------

Mapping:

Step 2:
Mapping of weak entity types
(None)

Step 3:
Mapping of 1:1 relationship type

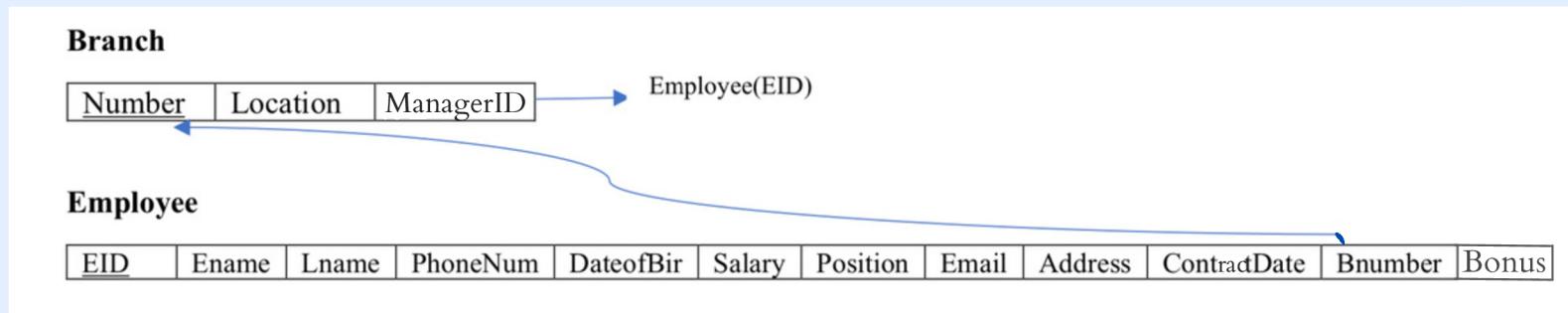
Relationship: manages



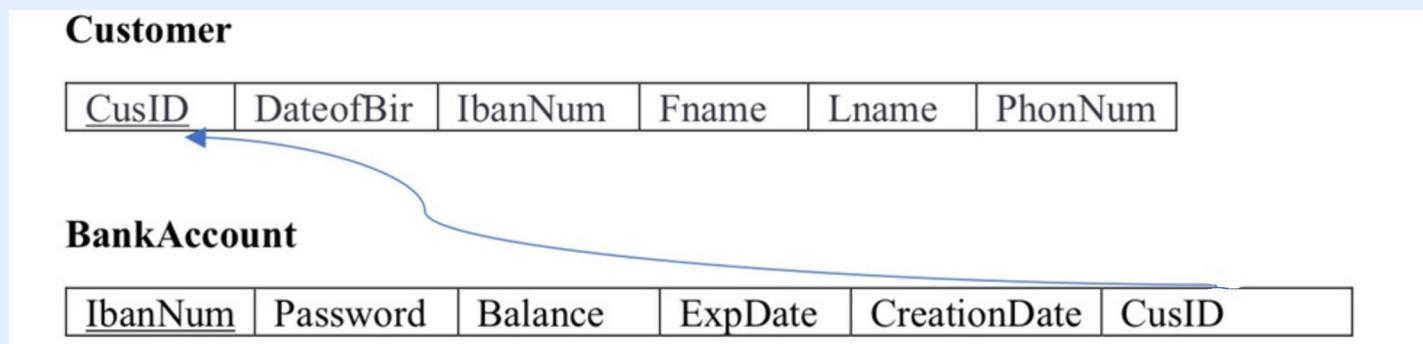
Mapping:

Step 4:
Mapping of 1:N relationship type

Relationship: works for



Relationship: owns



Mapping:

Step 5:

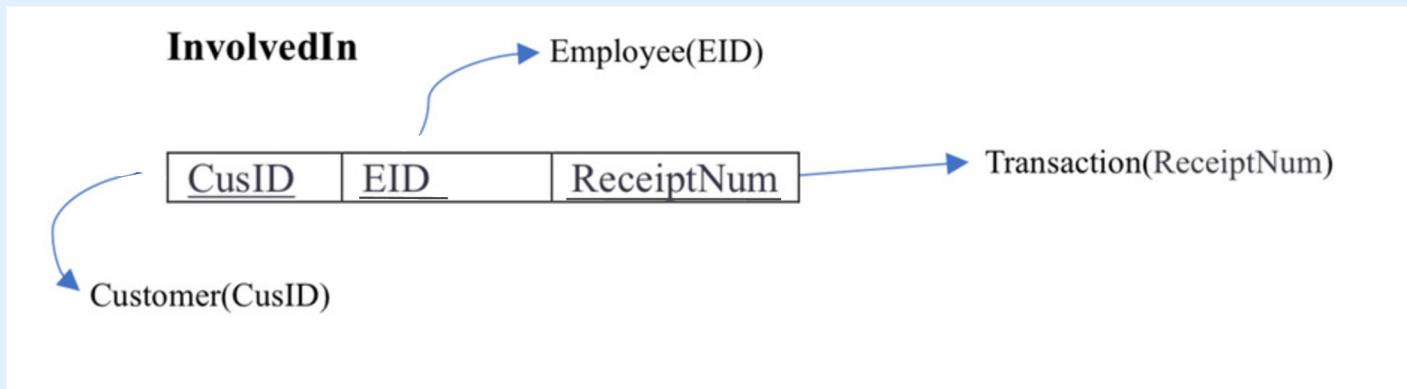
Mapping of M:N relationship type
(None)

Step 6:

Mapping of multi-valued attribute
(None)

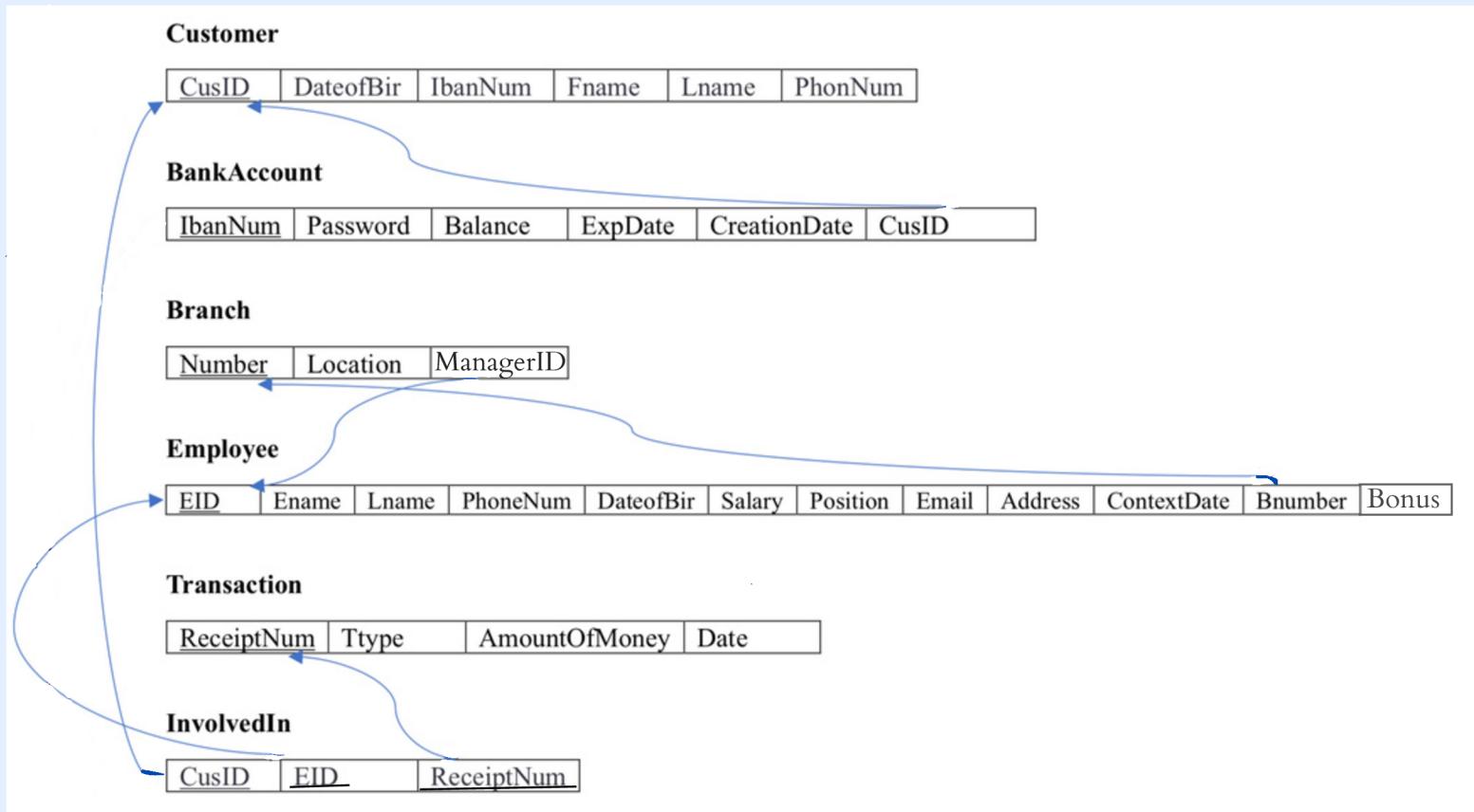
Step 7:

Mapping of N-ary relationship types



Mapping:

Final relational schema:



Database normalization:

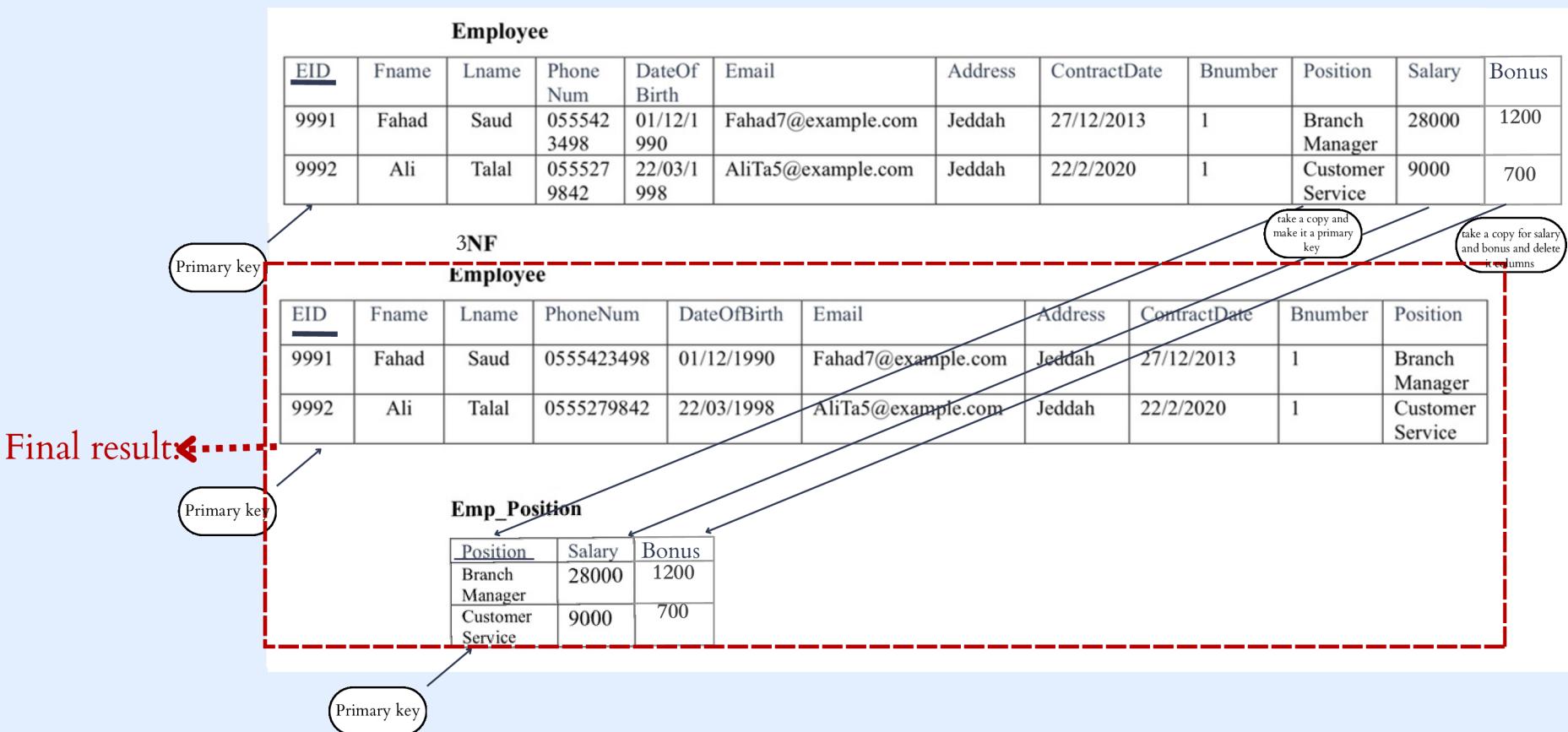
this table is in:

1NF(because there is no multi-valued attribute)

2NF(because there is no partial dependency)

But it is not in:

3NF(because the position is a none primary key attribute that identifies salary and bonus and they depend on it)



Database normalization:

They are already normalized

they are all in:

1NF(because there is no multi-valued attribute)

2NF(because there is no partial dependency)

3NF(because there is no transitive dependency)

Customer

<u>CustID</u>	DateOfBir	IbanNum	Fname	Lname	PhoneNum
334	01/9/1993	1122336790	Nora	Mohammed	541540565
335	04/9/2001	1122336791	Farah	Ali	555999456

BankAccount

<u>IbanNum</u>	Password	Balance	ExpDate	CreationDate	<u>CustId</u>
1122336790	2209	30654.78	01/6/2024	01/6/2020	334
1122336791	2003	300.00	03/6/2025	03/3/2021	335
1122336792	2009	0.00	20/5/2027	20/5/2023	334

More tables



Database normalization:

Branch

<u>BNumber</u>	location	MangId
1	jeddah	9991
2	makkah	9002

Transaction

<u>ReceiptNum</u>	TType	Date	AmountOfMoney
----	----	00/00/0000	00
----	----	00/00/0000	00

InolvedIn

<u>CustId</u>	<u>EId</u>	<u>ReceiptNum</u>

SQL tables and commands:

Customer table:

Creating & inserting the table:

```
356
357 • CREATE TABLE Customer (
358     CusID INT (4) PRIMARY KEY,
359     DateofBir DATE,
360     IbanNum VARCHAR(30),
361     Fname VARCHAR(20),
362     Lname VARCHAR(20),
363     PhonNum VARCHAR(13)
364 );
365
366 • INSERT INTO Customer
367
368     VALUES
369     (334, '1995-02-15', 'SA77 8000 0246 6080 1152 5869', 'Noor', 'Al-Sayed', '+966501234567'),
370     (335, '1996-04-10', 'SA68 8000 0246 6080 1367 8635', 'Layla', 'Al-Farouk', '+966501234568'),
371     (336, '1994-12-12', 'SA54 8000 0246 6080 4528 0019', 'Hana', 'Ibn-Battuta', '+966501234569'),
372     (337, '1993-07-07', 'SA12 8000 0246 6080 6643 6329', 'Yasmeen', 'Al-Rashid', '+966501234570'),
373     (338, '1992-11-20', 'SA46 8000 0246 6080 7980 2276', 'Samar', 'Al-Ghazali', '+966501234571);
```

Updating & deleting commands:

```
365 -- 2 update for customer
366
367 • UPDATE Customer
368     SET PhonNum = '+966501234578'
369     WHERE CusID = 334;
370
371
372 • UPDATE Customer
373     SET Fname = 'Aisha', Lname = 'Al-Khalid'
374     WHERE CusID = 336;
375
376 -- 2 delete for customer
377
378 • DELETE FROM Customer
379     WHERE CusID = 338;
380
381 • DELETE FROM Customer
382     WHERE Lname = 'Al-Rashid';
```

Before updating:

Result Grid						
	CusID	DateofBir	IbanNum	Fname	Lname	PhonNum
▶	334	1995-02-15	SA77 8000 0246 6080 1152 5869	Noor	Al-Sayed	+966501234567
	335	1996-04-10	SA68 8000 0246 6080 1367 8635	Layla	Al-Farouk	+966501234568
	336	1994-12-12	SA54 8000 0246 6080 4528 0019	Hana	Ibn-Battuta	+966501234569
	337	1993-07-07	SA12 8000 0246 6080 6643 6329	Yasmeen	Al-Rashid	+966501234570
*	338	1992-11-20	SA46 8000 0246 6080 7980 2276	Samar	Al-Ghazali	+966501234571
*				HULL	HULL	HULL

After updating:

Result Grid						
	CusID	DateofBir	IbanNum	Fname	Lname	PhonNum
▶	334	1995-02-15	SA77 8000 0246 6080 1152 5869	Noor	Al-Sayed	+966501234578
	335	1996-04-10	SA68 8000 0246 6080 1367 8635	Layla	Al-Farouk	+966501234568
	336	1994-12-12	SA54 8000 0246 6080 4528 0019	Aisha	Al-Khalid	+966501234569
*				HULL	HULL	HULL

SQL tables and commands:

BankAccount table:

Creating & inserting the table:

```
182 •  create table `bankAccount`  
183  
184   ( ibanNum      varchar(30)not null,  
185     APassword    int(5),  
186     Balance      decimal(10,2),  
187     CreationDate date,  
188     ExpDate      date,  
189     CusID        int(4),  
190     CONSTRAINT bankAccount_PK primary key(ibanNum)  
191   );  
192  
193 •  INSERT INTO `bankAccount`  
194  
195   VALUES  ('SA77 8000 0246 6080 1152 5869',64378,4000.50,"2020-10-14","2024-10-01",334),  
196   ('SA68 8000 0246 6080 1367 8635',74890,500.00,"2019-12-03","2023-12-01",335),  
197   ('SA54 8000 0246 6080 4528 0019',93241,10000.00,"2021-05-08","2025-05-01",336),  
198   ('SA12 8000 0246 6080 6643 6329',88243,2305.30,"2021-02-11","2025-02-01",337),  
199   ('SA46 8000 0246 6080 7980 2276',62809,78600.00,"2023-07-09","2027-07-01",338);
```

Updating & deleting commands:

```
221   -- 2 update for bank account  
222  
223 •  UPDATE `bankAccount`  
224   set  Balance = 450.00  
225   where ibanNum = 'SA68 8000 0246 6080 1367 8635' ;  
226  
227 •  UPDATE `bankAccount`  
228  
229   set  APassword = 65509  
230  
231   where ibanNum = 'SA46 8000 0246 6080 7980 2276' ;  
232  
233   -- 2 delete for bank account  
234  
235 •  DELETE FROM `bankAccount`  
236   WHERE ExpDate = "2023-12-01";  
237  
238  
239  
240 •  DELETE FROM `bankAccount`  
241   WHERE ibanNum = 'SA77 8000 0246 6080 1152 5869' ;
```

Before updating:

Result Grid	Filter Rows:	Edit:		Export/Import:	
ibanNum	APassword	Balance	CreationDate	ExpDate	CusID
SA12 8000 0246 6080 6643 6329	88243	2305.30	2021-02-11	2025-02-01	337
SA46 8000 0246 6080 7980 2276	62809	78600.00	2023-07-09	2027-07-01	338
SA54 8000 0246 6080 4528 0019	93241	10000.00	2021-05-08	2025-05-01	336
SA68 8000 0246 6080 1367 8635	74890	500.00	2019-12-03	2023-12-01	335
SA77 8000 0246 6080 1152 5869	64378	4000.50	2020-10-14	2024-10-01	334
NULL	NULL	NULL	NULL	NULL	NULL

After updating:

Result Grid	Filter Rows:	Edit:		Export/Import:	
ibanNum	APassword	Balance	CreationDate	ExpDate	CusID
SA12 8000 0246 6080 6643 6329	88243	2305.30	2021-02-11	2025-02-01	337
SA46 8000 0246 6080 7980 2276	65509	78600.00	2023-07-09	2027-07-01	338
SA54 8000 0246 6080 4528 0019	93241	10000.00	2021-05-08	2025-05-01	336
NULL	NULL	NULL	NULL	NULL	NULL

IMPORTANT CONSTRAINT:

```
357
358      -- Adding the foreign key after creating the Customer table
359
360 • ALTER TABLE bankaccount
361     ADD
362     CONSTRAINT bankAccount_FK foreign key(CusID)REFERENCES Customer(CusID)ON DELETE CASCADE;
363
364
```

SQL tables and commands:

Branch table:

Creating & inserting the table:

```
51 • ① create table Branch (
52     Bnum      int(3) not null,
53     location  varchar(20),
54     mangID    int(9) unique,
55     primary key(Bnum),
56
57     constraint Branch_FK1 foreign key(mangID) references `employee` (eID) on delete set null
58 );
59 •  INSERT INTO Branch (Bnum, location, mangID)
60     VALUES
61         (1, 'Jeddah', 9993),
62         (2, 'Makkah', 9994),
63         (3, 'Riyadh', 9877),
64         (4, 'Taif', 9897),
65         (5, 'Jubail', 9827);
```

Updating & deleting commands:

```
77     -- 2 update for Branch
78
79
80 •  UPDATE branch
81     SET
82         location = 'Alkhobr'
83     where
84         Bnum = 5;
85 •  UPDATE branch
86     SET
87         location = 'Abha'
88     where
89         Bnum = 4;
90
91
92     -- 2 delete for Branch
93
94 •  DELETE FROM branch
95     WHERE Bnum = 3 ;
96
97 •  DELETE FROM branch
98     WHERE Bnum = 4 ;
99
100 •  DELETE FROM branch
101    WHERE Bnum = 5;
```

Before updating:

	Bnum	location	mangID
▶	1	Jeddah	9993
	2	Makkah	9994
	3	Riyadh	9877
	4	Taif	9897
	5	Jubail	9827
*	NULL	NULL	NULL

After updating:

	Bnum	location	mangID
▶	1	Jeddah	9993
	2	Makkah	9994
*	NULL	NULL	NULL

SQL tables and commands:

Employee table:

Creating & inserting the table:

```
24 • ⚡ create table employee (
25     eID      int(4) not null ,
26     fName    varchar(20),
27     lName   varchar(20),
28     email   varchar(200)unique,
29     phoneNum int(10)unique,
30     DateOfBirth date,
31     address  varchar(20),
32     contactDate date,
33     Bnum     int(3),
34     position  varchar(40),
35
36     primary key(eID),
37     constraint employee_FK1 foreign key(position) references `emp_position` (position) on delete set null
38 );
39 •     INSERT INTO employee (Bnum, contactDate, address, email, DateOfBirth, phoneNum, lName, fName, eID, position)
40 VALUES
41     (1, '2013-12-09', 'Jeddah', 'Fahad7@example.com', '1998-12-01', 555423498, 'Saud', 'Fahad', 9991,'Bank Teller'),
42     (1, '2018-02-04', 'Jeddah', 'AliTalal17@example.com', '1998-03-22', 555279842, 'Talal', 'Ali', 9992, 'Loan Officer'),
43     (1, '2014-06-05', 'Jeddah', 'Ahmad7@example.com', '1994-02-09', 543269765, 'Sami', 'Ahmad', 9993, 'Branch Manager'),
44     (2, '2020-09-04', 'makkah', 'fisal33@example.com', '2003-09-04', 541540565 , 'Saad', 'Fisal', 9994, 'Branch Manager'),
45     (2, '2022-07-07', 'makkah', 'talal@example.com', '1993-09-01', 555999456 , 'ali', 'Talal', 9995, 'Customer Service Representative'),
46     (3, '2014-06-05', 'Riyadh', 'rasha@example.com', '1994-02-09', 543779765, 'Sami', 'Rasha', 9877, 'Branch Manager'),
47     (4, '2020-09-04', 'Taif', 'farah133@example.com', '2003-09-04', 533540565 , 'Saad', 'Farah', 9897, 'Branch Manager'),
48     (5, '2020-09-04', 'Jubail', 'sarraal33@example.com', '2003-09-04', 533549866 , 'Saad', 'Sara', 9827, 'Branch Manager');
```

Updating & deleting commands:

```
106          -- 2 delete for employee
107 •   DELETE FROM employee
108      WHERE EId = 9991;
109
110 •   DELETE FROM employee
111      WHERE EId = 9992;
112
113          -- 2 update for employee
114
115 •   UPDATE employee
116      SET
117          email = 'fisa3@example.com'
118      where
119          eID= 9994;
120
121
122 •   UPDATE employee
123      SET
124          position = 'Loan Officer'
125      where
126          eID= 9995;
```

Before updating:

After updating:

IMPORTANT CONSTRAINT:

```
69      -- Adding the foreign key after creating the branches table
70
71 • ALTER TABLE employee
72   ADD
73     constraint employee_FK2 foreign key(Bnum) references `Branch` (Bnum) on delete cascade;
74
75
```

SQL tables and commands:

Emp_position table:

Creating &inserting the table:

```
3 • ④ create table emp_position (
4     position      varchar(40) not null,
5     salary        int(10),
6     bonus         int(10),
7     primary key(position) );
8
9
10 • INSERT INTO emp_position (Position, Bonus, Salary)
11 VALUES
12     ('Bank Teller', 1000, 15000),
13     ('Loan Officer', 1500, 18000),
14     ('Branch Manager', 2000, 25000),
15     ('Financial Analyst', 2500, 30000),
16     ('Customer Service Representative', 1200, 16000);
```

Updating &deleting commands:

```
129      -- 2 delete for emp_position
130
131 •  DELETE FROM emp_position
132 WHERE Position IN ('Bank Teller');
133
134
135 •  DELETE FROM emp_position
136 WHERE Position IN ('Financial Analyst');
137
138
139
140      -- 2 update for emp_position
141
142 •  UPDATE emp_position
143     SET
144         Salary = 19000
145     WHERE Position = 'Loan Officer';
146
147 •  UPDATE emp_position
148     SET
149         Salary = 30000
150     WHERE Position = 'Branch Manager';
```

Before updating:

Result Grid Filter Rows:			
	position	salary	bonus
▶	Bank Teller	15000	1000
	Branch Manager	25000	2000
	Customer Service Representative	16000	1200
	Financial Analyst	30000	2500
	Loan Officer	18000	1500
*	NULL	NULL	NULL

After updating:

Result Grid Filter Rows:			
	position	salary	bonus
▶	Branch Manager	30000	2000
	Customer Service Representative	16000	1200
	Loan Officer	19000	1500
*	NULL	NULL	NULL

SQL tables and commands:

Transaction table:

Creating & inserting the table:

```
204 •  create table `Transaction`  
205  
206     ( ReceiptNum      int(6)not null,  
207         Ttype          varchar(15) CHECK (Ttype IN('deposit','withdrawls','transfer','online payment')),  
208         AmountOfMoney   decimal(7,2),  
209         Tdate          date,  
210         CONSTRAINT Transaction_PK primary key(ReceiptNum)  
211     );  
212  
213  
214 •  INSERT INTO `Transaction`  
215  
216     VALUES (217458,'transfer',500.00,"2023-05-17"),  
217     (217474,'withdrawls',200.00,"2023-06-11"),  
218     (217567,'online payment',350.00,"2023-07-23"),  
219     (217575,'deposit',750.00,"2023-07-27"),  
220     (217589,'online payment',220.00,"2023-08-01");
```

Updating & deleting commands:

```
276 -- 2 update for Transaction  
277  
278  
279 •  UPDATE `Transaction`  
280     set    AmountOfMoney = 315.00  
281     where  ReceiptNum  = 217567;  
282  
283  
284  
285 •  UPDATE `Transaction`  
286     set    Ttype = 'online payment'  
287     where  ReceiptNum  = 217458;  
288  
289  
290 -- 2 delete for Transaction  
291  
292 •  DELETE FROM `Transaction`  
293     WHERE  AmountOfMoney <= 200 ;  
294  
295  
296 •  DELETE FROM `Transaction`  
297     WHERE  Ttype = 'deposit';
```

Before updating:

Result Grid			
Filter Rows: <input type="text"/> Edit:			
ReceiptNum	Ttype	AmountOfMoney	Tdate
217458	transfer	500.00	2023-05-17
217474	withdrawls	200.00	2023-06-11
217567	online payment	350.00	2023-07-23
217575	deposit	750.00	2023-07-27
217589	online payment	220.00	2023-08-01
*	NULL	NULL	NULL

After updating:

Result Grid			
Filter Rows: <input type="text"/> Edit:			
ReceiptNum	Ttype	AmountOfMoney	Tdate
217458	online payment	500.00	2023-05-17
217567	online payment	315.00	2023-07-23
217589	online payment	220.00	2023-08-01
*	NULL	NULL	NULL

SQL tables and commands:

InvolvedIn table:

Can't be updated individually because it's an N-ary relationship and all the attributes are foreign keys (primary keys in another tables) so if the parent tables are updated this table will be updated too.

Creating & inserting the table:

```
459 • CREATE TABLE InvolvedIn (
460
461     CusID INT(4),
462     EID INT(4),
463     ReceiptNum INT(6),
464     CONSTRAINT InvolvedIn_FK1 FOREIGN KEY (CusID) REFERENCES customer(CusID)ON DELETE CASCADE,
465     CONSTRAINT InvolvedIn_FK2 FOREIGN KEY (EID) REFERENCES employee(eID)ON DELETE CASCADE,
466     CONSTRAINT InvolvedIn_FK3 FOREIGN KEY (ReceiptNum) REFERENCES `Transaction`(ReceiptNum)ON DELETE CASCADE
467
468 );
469 • INSERT INTO InvolvedIn
470
471     VALUES
472     (336, 9993, 217567),
473     (337, 9994, 217575),
474     (338, 9995, 217589);
```

SELECT commands:

1-Using ORDER BY:

```
450 • SELECT CusID,DateofBir,IbanNum,Fname  
451  
452     FROM Customer  
453  
454     ORDER BY CusID;  
455  
456  
457  
458  
459     -- (8) where  
460 • SELECT CusID,Fname,Lname,PhonNum
```

Result Grid | Filter Rows: | Edit: | | | | | |

CusID	DateofBir	IbanNum	Fname
334	1995-02-15	SA77 8000 0246 6080 1152 5869	Noor
335	1996-04-10	SA68 8000 0246 6080 1367 8635	Layla
336	1994-12-12	SAS4 8000 0246 6080 4528 0019	Aisha
*	NULL	NULL	NULL

2-Using WHERE:

```
460 • SELECT CusID,Fname,Lname,PhonNum  
461  
462     FROM Customer  
463  
464     WHERE Fname='Noor';  
465  
466  
467  
468  
469
```

Result Grid | Filter Rows: | Edit: | | | | | |

CusID	Fname	Lname	PhonNum
334	Noor	Al-Sayed	+966501234578
*	NULL	NULL	NULL

SELECT commands:

3-Using join operations-right :

```
481 • SELECT *
482   FROM customer AS L
483     RIGHT JOIN bankaccount AS R
484   ON L.CusID = R.CusID;
485
486
487
```

Result Grid												
CusID	DateofBir	IbanNum	Fname	Lname	PhonNum	IbanNum	APassword	Balance	CreationDate	ExpDate	CusID	
336	1994-12-12	SAS4 8000 0246 6080 4528 0019	Aisha	Al-Khalid	+966501234569	SAS4 8000 0246 6080 4528 0019	93241	10000.00	2021-05-08	2025-05-01	336	

4-Using Subquerie&WHERE:

```
278 • SELECT *
279
280   FROM `bankAccount`
281
282   WHERE ibanNum = (SELECT ibanNum
283
284     FROM `bankAccount`
285
286     WHERE APassword = 93241);
287
288
289   -- (4) Order by
```

Result Grid						
IbanNum	APassword	Balance	CreationDate	ExpDate	CusID	
SAS4 8000 0246 6080 4528 0019	93241	10000.00	2021-05-08	2025-05-01	336	
*	NULL	NULL	NULL	NULL	NULL	

SELECT commands:

5-Using ORDER BY:

```
290 •   SELECT CusID,CreationDate,Balance  
291  
292     from `bankAccount`  
293  
294     ORDER BY CreationDate ;  
295  
296  
297  
298 •   CREATE VIEW card as
```

Result Grid | Filter Rows: Export:

CusID	CreationDate	Balance
336	2021-05-08	10000.00

6-Using join operations-left:

```
168  
169 •   SELECT *  
170     FROM Branch as l  
171       LEFT JOIN employee as r  
172         ON l.mangID = r.eID;  
173
```

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

Bnum	location	mangID	eID	fName	lName	email	phoneNum	DateOfBirth	address	contactDate	Bnum	position
1	Jeddah	9993	9993	Ahmad	Sami	Ahmad7@example.com	543269765	1994-02-09	Jeddah	2014-06-05	1	Branch Manager
2	Makkah	9994	9994	Fisal	Saad	fisa3@example.com	541540565	2003-09-04	makkah	2020-09-04	2	Branch Manager

Result Grid

SELECT commands:

7-Using join operations&WHERE:

```
89 •  SELECT SUM(emp_position.salary) + SUM(emp_position.bonus) as total_salary_bonus
90   FROM employee
91   JOIN emp_position ON employee.position = emp_position.position
92   WHERE employee.Bnum = 1 ;
93
```

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

CusID	DateofBir	IbanNum	Fname	Lname	PhonNum	ibanNum	APassword	Balance	CreationDate	ExpDate	CusID
336	1994-12-12	SA54 8000 0246 6080 4528 0019	Aisha	Al-Khalid	+966501234569	SA54 8000 0246 6080 4528 0019	93241	10000.00	2021-05-08	2025-05-01	336

8-Using subquerie&GROUP BY:

```
177 •  SELECT
178   position,
179   (SELECT AVG(salary)
180    FROM emp_position
181    WHERE emp_position.position = employee.position) AS average_salary
182   FROM employee
183   GROUP BY position;
184
185
```

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

position	average_salary
Branch Manager	30000.0000
Loan Officer	19000.0000

SELECT commands:

9-Using GROUP BY&HAVING:

```
349
350 •   SELECT Ttype , count(ReceiptNum) AS numberOfTransactions, SUM(AmountOfMoney) AS moneySum
351
352     from `Transaction`
353
354     GROUP BY Ttype
355
356     HAVING Ttype = 'online payment';
357
358
```

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

Ttype	numberOfTransactions	moneySum
online payment	3	1035.00

10-Using WHERE&ORDER BY:

```
340 •   SELECT ReceiptNum,Ttype,AmountOfMoney,Tdate
341
342     from `Transaction`
343
344     WHERE AmountOfMoney > 350.00
345
346     ORDER BY 3 ;
347
348     -- (6) group by and Having
349
```

Result Grid | Filter Rows: | Edit: |

ReceiptNum	Ttype	AmountOfMoney	Tdate
217458	online payment	500.00	2023-05-17
NULL	NULL	NULL	NULL