Al-Imam University
Computer Science Department
College of Computer and Information Sciences
1st Semester 1443-1444 H

CS370: Database

Computer Online Auction Store

(Project Report)

Section Code: 373.
Due date: 14th Dec 2021

Group name: Group1

Group's Members Info:

Student Name	Student Id	Email
Noor Almasry	440027219	nelmasry@sm.imamu.edu.sa
Norah Alsharhan	440022381	nalsharhan@sm.imamu.edu.sa
Raghad Albosais	440020209	rkaalbosais@sm.imamu.edu.sa

Group Leader: Raghad Albosais

Content Table

Figures Table

1. The goal of this project

This project aims to build a DB for an online website that make auctions on used items, and offers a lot of facilities to give the client the best service they need to buy and sell.

2. Project description

Consider a computer online auction store where clients (purchasers and suppliers) participate in the sales process for the electronic items. the requirements are summarized as follows to build the database for eHaraj:

• The online site has Clients, each of whom is identified by a unique client id and is also described by an email, name which consist of first and last name, password, client address and phone numbers; where each client may add more than one phone numbers.

▪ All the clients' attributes cannot be null in the database.

• A client should be either: a purchaser (i.e. who buy items), or a supplier (i.e. who sell items).

- A purchaser has a billing address in addition.

- A supplier has a bank account number in addition.

▪ Billing address cannot be null.

▪ bank account number cannot be null, and need to be checked that account number length is= 10

• The database also stores the Items, items are placed and owned by a Supplier, and each item in database should belong to one supplier. Items are identified by a unique item id, also items described by an item name (i.e. product name), item manufacturer name, items languages (i.e. means that the languages that the computer or printer provide, such as Arabic, English ..etc), items used months (i.e. means that how many months this item been used by supplier, since it is auction and the items may not be new), a description, starting bid price, the start date of the auction, and the end date of the auction.

▪ All the items' attributes cannot be null in the database.

▪ The end date of the auction should be checked to be >= start date of the auction

• An Items should be either: Computer, or Printer. - A Computer have in addition a speed, memory, and storage.

- A Printer have in addition is_wireless , Inkjet and is_contain_scanner.

▪ All the attributes of computer and printer cannot be null.

• Purchaser (i.e., the bidder) may make one or more bids, and those bids are identified by Bid id, and also described by, proposed price (i.e. the price the purchaser proposed to buy the item), and bid date. Each bid proposed by one and only one purchaser.

▪ All the attributes of bid cannot be null.

• Bids are done on the items; each item may have one to many bids. Each bid assigned to one and only one item (i.e., means that the bid done on one specific item).

• Each Bid has a billing. A billing is identified by a unique bill id, also described by billing date. Each bill should assign to one and only one bid, also each bid may assign to only one bill.

▪ All the attributes of billing cannot be null.

▪ (Note to be considered in phase 2: only the purchaser with winner bid of an item higher price will have a bill, and if the purchaser wins in many items; each bid will have separate bill)

• The client may record one to many feedbacks. The feedback is identified and attached to both client and billing. Each feedback written by one and only one client, also written on one and only one bill. Having feedback is optional. Feedback is identified by feedback description and also described by a rating.

▪ Rating should be checked to be ranged between 1 and 5.

▪ feedback description cannot be null.

3. Conceptual EER Model

3.1. Data modeling tool

Name: Drow.io

Reference: https://app.diagrams.net/

3.2. Assumptions

1- Bank account number of the supplier is unique.

2- The supplier may placed and owned one or more item.

3- Each billing may have one or more feedback written on it.

4- The feedback is written by one the client.

5- The items can have multiple languages (Arabic and English).

6- The clients may have multiple phone numbers.

3.3. Description of EER data model component

1- Entity and attribute

- A client entity(superclass) should be either a purchaser entity(subclass) or a supplier entity(subclass), but not both. So, we represented purchaser and supplier as a

specialization of client with total and disjoint specialization type. Each client has primary key which is client_id, and has set of attributes which are an e-mail, password, client address, phone numbers is multivalued attribute, and name is composed attribute that consist of first and last name attributes. Purchaser entity is subclass has one additional attribute that does not exist in supplier, billing address. And supplier entity has one additional unique attribute that not exsit in purchaser, bank account number.

- Item entity(superclass) should be computer entity(subclass) or printer(subclass) but not both. So we represented computer and printer as a specialization of Item with total and disjoint specialization type. Computer entity is subclass has many additional attributes that do not exist in printer, which are speed, memory, and storage. Printer entity is subclass has many additional attributes that do not exist in hardware, which are is_wireless, is_contain_scanner and inkjet.
- Billing is entity and has primary key attribute billing_id, and another attribute which is billing date.
- Feedback is weak entity related to client identifying entity and billing identifying entity, it has partial key attribute feedback_description, and another attribute which is rating.
- Bid is entity and has primary key attribute bid_id , and another attributes which are bid_date and proposed_price.

2-Relationship types and constrains

- Own: each supplier may placed and owned one or more items in the store and each item must be placed and owned by only one supplier
- Written_by: each feedback should be written by only one client, and each client may write one or more feedbacks.
- Written_on: each feedback should be written on only one billing, and each billing may written on it one or more feedback.
- Assign: each billing should be assigned to only one bid, and each bid may be assigned to only one billing.
- Make: each purchaser may make one or more bid, and each bid should be made and proposed by only one purchaser.
- Done: each bid should be done on only one item, and each item may have one or more bids.

3.4. Conceptual data model

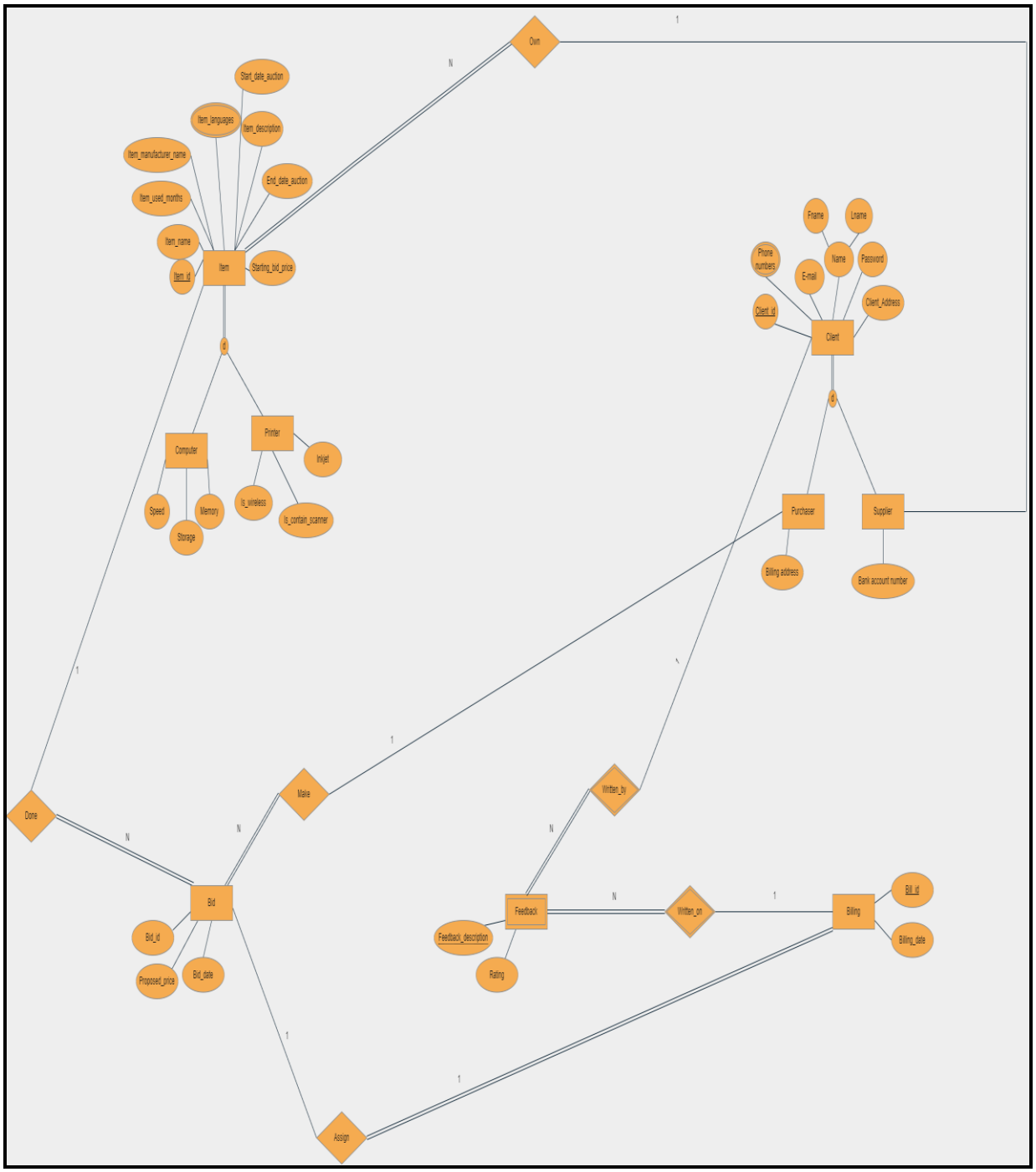


Figure 1: EER diagram

4. Relational database schema

4.1. Assumptions

There is no assumption.

4.2. Relational database schema

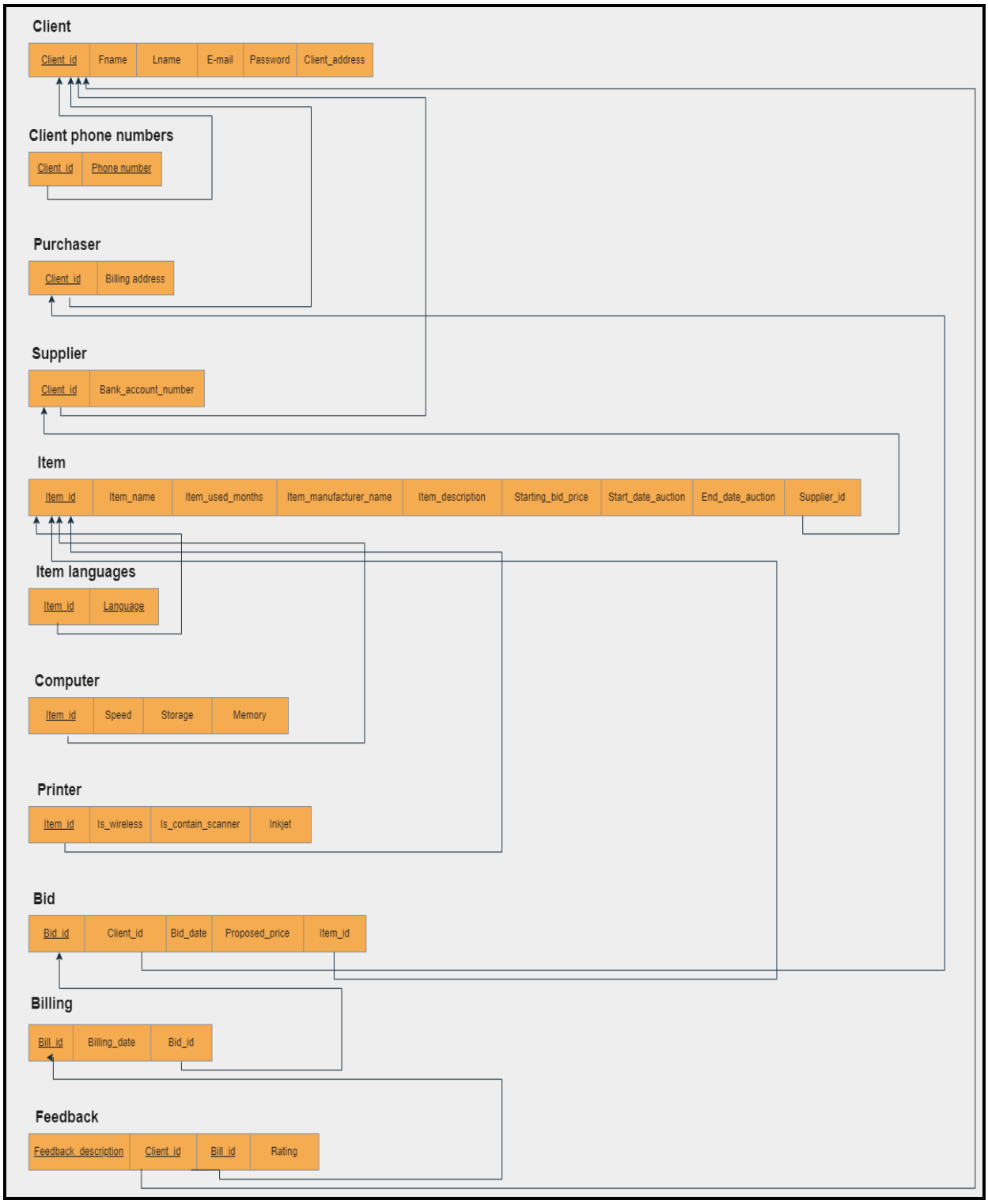


Figure 2: Relational model

5. Implementation DDL and DML

5.1. Define database

- Client table

```
create table Client (

Client_id  char(8) not null,
Fname    varchar(15) not null,
Lname    varchar(15) not null,
Email   varchar(30) not null,
Client_password varchar(20) not null check
(length(Client_password) >=8 ),
Client_address  varchar(50) not null,

primary key (Client_id));
```

- Client Phone numbers table

```
create table Client_phone_numbers (

Client_id char(8) not null,
Phone_number char(10) not null,

primary key (Client_id, Phone_number),
unique (Phone_number),
foreign key (Client_id) references Client(Client_id) on delete set
null);
```

- Supplier table

```
create table Supplier (

Client_id  char(8) not null,
Bank_account_number char(10) not null check (length(Bank_account_number)
= 10),
primary key (Client_id),
unique (Bank_account_number),
foreign key (Client_id) references Client(Client_id));
```

- Purchaser table

```
create table Purchaser (

Client_id  char(8) not null,
Billing_address varchar(50) not null,
primary key (Client_id),
foreign key (Client_id) references Client(Client_id));
```

- Item table

```
create table Item (

Item_id char(8) not null,
Item_name varchar(15) not null,
Item_used_months int not null,
Item_manufacturer_name varchar(15) not null,
Item_description varchar(50) not null,
Starting_bid_price float not null,
Start_date_auction date not null,
End_date_auction date not null,
Supplier_id char(8) not null,

primary key (Item_id),
foreign key (Supplier_id) references Supplier(Client_id),
constraint CHK_end_date check (Start_date_auction <= End_date_auction));
```

- Item language table

```
create table Item_languages (

Item_id char(8) not null,
Language varchar(7) not null check(Language = 'Arabic' or Language =
'English'),

primary key (Item_id,Language),
foreign key (Item_id) references Item(Item_id));
```

- Printer table

```
create table Printer (

Item_id char(8) not null,
Is_wireless varchar(3) not null check(Is_wireless='Yes' or
Is_wireless='No'),
Is_contain_scanner varchar(3) not null check(Is_contain_scanner='Yes'
or Is_contain_scanner='No'),
Inkjet varchar(3) not null check(Inkjet='Yes' or Inkjet='No'),

primary key (Item_id),
foreign key (Item_id) references Item(Item_id) on delete cascade
enable);
```

- Computer table

```
create table Computer (

Item_id char(8) not null,
Speed varchar(7) not null,
Storage_device varchar(3) not null check(Storage_device = 'HDD' or
Storage_device = 'SSD'),
RAM varchar(4) not null,

primary key (Item_id),
foreign key (Item_id) references Item(Item_id) );
```

- Bid table

```
create table Bid (

Bid_id char(8) not null,
Client_id char(8) not null,
Bid_date date not null,
Proposed_price float not null,
Item_id char(8) not null,
primary key (Bid_id) ,
foreign key (Client_id) references Purchaser(Client_id),
foreign key (Item_id) references Item(Item_id) on delete cascade
enable);
```

- Billing table

```
create table Billing (

Bill_id char(8) not null,
Billing_date date not null,
Bid_id char(8) not null,

primary key (Bill_id),
foreign key (Bid_id) references Bid(Bid_id));
```

- Feedback table

```
create table Feedback (

Feedback_description varchar(100) not null,
Client_id char(8) not null,
Bill_id char(8) not null,
Rating int check (Rating>=1 and Rating<=5),


primary key (Feedback_description, Client_id, Bill_id),
foreign key (Bill_id) references Billing(Bill_id),
foreign key (Client_id) references Client(Client_id));
```

5.2. Referential integrity constraints

Table	Referencing table(attribute)	Referenced table(attribute)	On DELETE action	On UPDATE action
Client Phone numbers	Client Phone numbers(Client_id)	Client(Client_id)	SET NULL	NO ACTION
Printer	Printer(Item_id)	Item(Item_id)	CASCADE	NO ACTION
Bid	Bid(Client_id) Bid(Item_id)	PurchaserClient_id) Item(Item_id)	NO ACTION CASCADE	NO ACTION NO ACTION

5.3. Populate database

- Insert client data

```
insert into Client values
  ('88866555', 'Raghad', 'Albosais', 'raghadkhaled_24@hotmail.com',
'Zce*]Y94m+','9999-Al ezdihar-Riyadh' );

insert into Client values
  ('33344555', 'Norah', 'Alsharhan', 'norah.ahammed@gmail.com',
'#p2t>8zHm8', '8756-AL olaya-Riyadh');

insert into Client values
  ('98765432', 'Noor', 'Almasry', 'noor.hossam.elmasry@gmail.com',
'46Y]G!hsj.', '1111-Al Almaseef-Riyadh');

insert into Client values
  ('11111110', 'Khaled', 'Alqhatani', 'khaled_alqh@gmail.com',
'}8vDFN<=e;', '0101-Al falah-Dubai');

insert into Client values
  ('44444440', 'Osama', 'Alsaeed', 'O_alsaeed@gmail.com', 'C85w*$()FN',
'2229-Al malaz-Riyadh');

insert into Client values
  ('55555550', 'Sara', 'Alshahrani', 'soso_1234@hotmail.com',
'\eny>32Str', '4863-Al nozha-Dahran');

insert into Client values
  ('12345678', 'Ali', 'Almutairi', 'Ali_Almutairi@hotmail.com',
'Jeoc@mkf56', '4863-Al Alnaseem-Riyadh');

insert into Client values
  ('01010100', 'Malak', 'Albaraak', 'baraak_malak@hotmail.com',
'Ihgy\nhkm34', '4863-Al Alnuzha-Dammam');

insert into Client values
  ('12654312', 'Alaa', 'Alfaise', 'AFaise@hotmail.com', 'Cvbn<hd234',
```

```
'4863-Al Alhamraa-Jeddah');

insert into Client values
  ('45673883', 'hoor', 'Almutairi', 'AlmutairiHOOR@hotmail.com',
'Jfjgin<r4', '4863-Al Alzohour-Dammam');

insert into Client values
  ('98549854', 'Tariq', 'Alzahrani', 'tariqqqq.zzz@hotmail.com',
'Djirwm@2357', '4863-Al Almurjan-jeddah');

insert into Client values
  ('34433443', 'Najla', 'Alotaybi', 'Najollaa_Otaybii@hotmail.com',
'Ubdnkw<dh64', '4863-Al Alnaseem-Dammam');
--------------------------------------------------------------------------------
```

CLIENT_ID	FNAME	LNAME	EMAIL	CLIENT_PASSWORD	CLIENT_ADDRESS
88866555	Raghad	Albosais	raghadkhaled_24@hotmail.com	Zce*]Y94m+	9999-Al ezdihar-Riyadh
33344555	Norah	Alsharhan	norah.ahammed@gmail.com	#p2t>8zHm8	8756-AL olaya-Riyadh
98765432	Noor	Almasry	noor.hossam.elmasry@gmail.com	46Y]G!hsj.	1111-Al Almaseef-Riyadh
11111110	Khaled	Alqhatani	khaled_alqh@gmail.com	}8vDFN<=e;	0101-Al falah-Dubai
44444440	Osama	Alsaeed	O_alsaeed@gmail.com	C85w*$()FN	2229-Al malaz-Riyadh
55555550	Sara	Alshahrani	soso_1234@hotmail.com	\eny>32Str	4863-Al nozha-Dahran
12345678	Ali	Almutairi	Ali_Almutairi@hotmail.com	Jeoc@mkf56	4863-Al Alnaseem-Riyadh
01010100	Malak	Albaraak	baraak_malak@hotmail.com	Ihgy\nhkm34	4863-Al Alnuzha-Dammam
12654312	Alaa	Alfaise	AFaise@hotmail.com	Cvbn<hd234	4863-Al Alhamraa-Jeddah
45673883	hoor	Almutairi	AlmutairiHOOR@hotmail.com	Jfjgin<r4	4863-Al Alzohour-Dammam
98549854	Tariq	Alzahrani	tariqqqq.zzz@hotmail.com	Djirwm@2357	4863-Al Almurjan-jeddah
34433443	Najla	Alotaybi	Najollaa_Otaybii@hotmail.com	Ubdnkw<dh64	4863-Al Alnaseem-Dammam

Figure 3: client table after insertion

- Insert client phone numbers data

```
insert into Client_phone_numbers values
  ('88866555', '0598365777');

insert into Client_phone_numbers values
  ('88866555', '0598369999');
```

```
insert into Client_phone_numbers values
  ('33344555','0505877231');

insert into Client_phone_numbers values
  ('98765432', '0544218854');

insert into Client_phone_numbers values
  ('12345678', '0508883322');

insert into Client_phone_numbers values
  ('12345678', '0508884141');

insert into Client_phone_numbers values
  ('01010100', '0500992311');

insert into Client_phone_numbers values
  ('12654312', '0544829513');
```

CLIENT_ID	PHONE_NUMBER
01010100	0500992311
12345678	0508883322
12345678	0508884141
12654312	0544829513
33344555	0505877231
88866555	0598365777
88866555	0598369999
98765432	0544218854

Figure 4: phone numbers table after insertion

- Insert supplier data

```
insert into Supplier values
  ('12345678', '1234567890');

insert into Supplier values
  ('01010100', '0987654321');

insert into Supplier values
```

```
  ('12654312', '6543210987');


insert into Supplier values
  ('45673883', '2224448885');

insert into Supplier values
  ('98549854', '2345234523');

insert into Supplier values
  ('34433443', '7766554433');
```

CLIENT_ID	BANK_ACCOUNT_NUMBER
12345678	1234567890
01010100	0987654321
12654312	6543210987
45673883	2224448885
98549854	2345234523
34433443	7766554433

Figure 5: supplier table after insertion

- Insert purchaser data

```
insert into Purchaser values
  ('88866555','8380-AL olaya-Riyadh');

insert into Purchaser values
  ('33344555','7760-AL yasmin-Riyadh');

insert into Purchaser values
  ('98765432','6520-AL ghadir-Riyadh');

insert into Purchaser values
  ('11111110','5560-AL waha-Riyadh');

insert into Purchaser values
```

```
  ('44444440','2330-AL swedi-Riyadh');

insert into Purchaser values
  ('55555550','4570-AL narjes-Riyadh');
```

CLIENT_ID	BILLING_ADDRESS
88866555	8380-AL olaya-Riyadh
33344555	7760-AL yasmin-Riyadh
98765432	6520-AL ghadir-Riyadh
11111110	5560-AL waha-Riyadh
44444440	2330-AL swedi-Riyadh
55555550	4570-AL narjes-Riyadh

Figure 6: purchaser table after insertion

- Insert Item data

```
insert into Item values
  ('13579135', 'MacBook_Pro', 3, 'Apple', 'Touch bar, intel core i7 6
core, 16", SSD', 6750.99, date '2021-08-12', date '2021-12-01',
'12345678');

insert into Item values
  ('24680246', 'HP pavilion', 6, 'HP', 'Gaming laptop, intel core i7-
11370H, 15.6", SSD/1', 3599.99, date '2021-10-02', date '2021-12-03',
'01010100');

insert into Item values
  ('09650965', 'Dell inspiron', 12, 'Dell', 'convertible folder, intel
core i7-1195G7,14",SSD', 2999.99, date '2020-12-20',date '2021-12-04',
'12654312');

insert into Item values
  ('10101011', 'MacBook_Air', 3, 'Apple', 'Gray , M1 8-core, 13.3",
SSD', 3999.99, date '2021-08-01', date '2021-08-02', '12345678');

insert into Item values
```

```
  ('22002200','MateBook', 1, 'Huawei', 'Grey, Intel Core i5, 14", HDD,
Wi-Fi/Bluetooth', 500.00, date '2021-12-01', date '2021-12-01',
'45673883');


insert into Item values
  ('33003300', 'Surface Go', 4, 'Microsoft', 'Platinum, Intel Core i5,
12.4", HDD', 1000.00, date '2021-12-03', date '2021-12-05','01010100');

insert into Item values
  ('99988877', 'Smart tank', 2, 'HP', 'All in one, copy, print, scan,
Wi-Fi, inkjet', 800.00, date '2021-06-20', date '2021-12-07',
'45673883');

insert into Item values
  ('56565656', 'EcoTank', 9, 'Epson', 'All in one, copy, print, scan,
Wi-Fi, inkjet', 1000.00, date '2021-09-20', date '2021-12-08',
'98549854');

insert into Item values
  ('11223344', 'Neverstop', 14, 'HP', 'Laser printer, 1000w, Wi-Fi,
Gray/White', 999.99, date '2021-01-06', date '2021-12-09', '34433443');

insert into Item values
  ('11100110', 'PIXMA', 2, 'Canon', 'All in one, copy, print, scan, Wi-
Fi, inkjet', 550.00, date '2021-04-13', date '2021-04-23', '34433443');

insert into Item values
  ('12345689', 'i-SENSYS', 8, 'canon', 'All in one, copy, print, scan,
Laser ', 450.00, date '2021-06-01', date '2021-06-01', '45673883');

insert into Item values
  ('21123223', 'EcoTank', 3, 'Epson', '  Inkjet printer, Wi-Fi, Black',
1944.79, date '2021-11-01', date '2021-11-01', '12654312');

-----------------------------------------------------------------------
```

ITEM_ID	ITEM_NAME	ITEM_USED_MONTHS	ITEM_MANUFACTURER_NAME	ITEM_DESCRIPTION	STARTING_BID_PRICE	START_DATE_AUCTION	END_DATE_AUCTION	SUPPLIER_ID
13579135	MacBook_Pro	3	Apple	Touch bar, intel core i7 6 core, 16", SSD	6750.99	12-AUG-21	01-DEC-21	12345678
24680246	HP pavilion	6	HP	Gaming laptop, intel core i7-11370H, 15.6", SSD/1	3599.99	02-OCT-21	03-DEC-21	01010100
09650965	Dell inspiron	12	Dell	convertible folder, intel core i7-1195G7,14",SSD	2999.99	20-DEC-20	04-DEC-21	12654312
10101011	MacBook_Air	3	Apple	Gray , M1 8-core, 13.3", SSD	3999.99	01-AUG-21	02-AUG-21	12345678
22002200	MateBook	1	Huawei	Grey, Intel Core i5, 14", HDD, Wi-Fi/Bluetooth	500	01-DEC-21	01-DEC-21	45673883
33003300	Surface Go	4	Microsoft	Platinum, Intel Core i5, 12.4", HDD	1000	03-DEC-21	05-DEC-21	01010100
99988877	Smart tank	2	HP	All in one, copy, print, scan, Wi-Fi, inkjet	800	20-JUN-21	07-DEC-21	45673883
56565656	EcoTank	9	Epson	All in one, copy, print, scan, Wi-Fi, inkjet	1000	20-SEP-21	08-DEC-21	98549854
11223344	Neverstop	14	HP	Laser printer, 1000w, Wi-Fi, Gray/White	999.99	06-JAN-21	09-DEC-21	34433443
11100110	PIXMA	2	Canon	All in one, copy, print, scan, Wi-Fi, inkjet	550	13-APR-21	23-APR-21	34433443
12345689	i-SENSYS	8	canon	All in one, copy, print, scan, Laser	450	01-JUN-21	01-JUN-21	45673883
21123223	EcoTank	3	Epson	Inkjet printer, Wi-Fi, Black	1944.79	01-NOV-21	01-NOV-21	12654312

Figure 7: Item table after insertion

- Insert Item languages data

```
insert into Item_languages values
  ( '13579135', 'Arabic');

insert into Item_languages values
  ( '13579135', 'English');

insert into Item_languages values
  ( '24680246', 'English');

insert into Item_languages values
  ( '09650965', 'English');

insert into Item_languages values
  ('10101011', 'English');

insert into Item_languages values
  ( '22002200', 'English');

insert into Item_languages values
  ('33003300', 'English');

insert into Item_languages values
  ('33003300', 'Arabic');
```

```
insert into Item_languages values
  ('99988877', 'English');

insert into Item_languages values
  ('99988877', 'Arabic');

insert into Item_languages values
  ('56565656', 'Arabic');

insert into Item_languages values
  ('11223344', 'English');

insert into Item_languages values
  ( '12345689', 'Arabic');

insert into Item_languages values
  ( '11100110', 'Arabic');

insert into Item_languages values
  ( '21123223', 'Arabic');

insert into Item_languages values
  ( '21123223', 'English');
```

ITEM_ID	LANGUAGE
09650965	English
10101011	English
11100110	Arabic
11223344	English
12345689	Arabic
13579135	Arabic
13579135	English
21123223	Arabic
21123223	English
22002200	English
24680246	English
33003300	Arabic
33003300	English
56565656	Arabic

Figure 8: Item language table after insertion

- Insert printer data

```
insert into Printer values
  ('99988877','Yes','Yes','Yes');

insert into Printer values
  ('56565656','Yes','Yes','Yes');

insert into Printer values
  ('11223344','Yes','Yes','No');

insert into Printer values
  ('11100110','Yes','Yes','Yes');

insert into Printer values
  ('12345689','Yes','Yes','No');

insert into Printer values
  ('21123223','Yes','Yes','Yes');

--------------------------------------------------------------------------
```

ITEM_ID	IS_WIRELESS	IS_CONTAIN_SCANNER	INKJET
99988877	Yes	Yes	Yes
56565656	Yes	Yes	Yes
11223344	Yes	Yes	No
11100110	Yes	Yes	Yes
12345689	Yes	Yes	No
21123223	Yes	Yes	Yes

Figure 9: printer table after insertion

- Insert computer data

```
insert into Computer values
  ('13579135','1.2 GHz','SSD','64GB');

insert into Computer values
```

```
  ('24680246','2.6 GHz','SSD','8GB');

insert into Computer values
  ('09650965','1.3 GHz','SSD','16GB');

insert into Computer values
  ('10101011','1.9 GHz','SSD','64GB');

insert into Computer values
  ('22002200','1.1 GHz','HDD','64GB');

insert into Computer values
  ('33003300','1.8 GHz','HDD','16GB');
```

ITEM_ID	SPEED	STORAGE_DEVICE	RAM
13579135	1.2 GHz	SSD	64GB
24680246	2.6 GHz	SSD	8GB
09650965	1.3 GHz	SSD	16GB
10101011	1.9 GHz	SSD	64GB
22002200	1.1 GHz	HDD	64GB
33003300	1.8 GHz	HDD	16GB

Figure 10: computer table after insertion

- Insert bid data

```
insert into Bid values
  ('11111111','88866555', date '2021-12-01', 7000.00,'13579135');

insert into Bid values
  ('22222222','33344555', date '2021-12-01', 8000.00,'13579135');

insert into Bid values
  ('33333333','98765432', date '2021-12-02', 4500.88,'24680246');

insert into Bid values
  ('44444444','98765432', date '2021-12-03', 5500.99,'24680246');
```

```
insert into Bid values
  ('55555555','11111110', date '2021-12-03', 4900.76,'24680246');

insert into Bid values
  ('66666666','11111110', date '2021-12-04',3100.87,'09650965');

insert into Bid values
  ('77777777','11111110', date '2021-12-04',3800.00 ,'09650965');

insert into Bid values
  ('88888888','88866555', date '2021-12-07', 960.88,'99988877');

insert into Bid values
  ('99999999','44444440', date '2021-12-07', 920.00, '99988877');

insert into Bid values
  ('10101010','44444440', date '2021-12-08', 1200.99,'56565656');

insert into Bid values
  ('12121212','33344555', date '2021-12-08', 1900.00,'56565656');

insert into Bid values
  ('13131313','55555550', date '2021-12-09', 1200.0,'11223344');

insert into Bid values
  ('14141414','55555550', date '2021-12-09', 1400.00,'56565656');
----------------------------------------------------------------------
```

BID_ID	CLIENT_ID	BID_DATE	PROPOSED_PRICE	ITEM_ID
11111111	88866555	01-DEC-21	7000	13579135
22222222	33344555	01-DEC-21	8000	13579135
33333333	98765432	02-DEC-21	4500.88	24680246
44444444	98765432	03-DEC-21	5500.99	24680246
55555555	11111110	03-DEC-21	4900.76	24680246
66666666	11111110	04-DEC-21	3100.87	09650965
77777777	11111110	04-DEC-21	3800	09650965
88888888	88866555	07-DEC-21	960.88	99988877
99999999	44444440	07-DEC-21	920	99988877
10101010	44444440	08-DEC-21	1200.99	56565656
12121212	33344555	08-DEC-21	1900	56565656
13131313	55555550	09-DEC-21	1200	11223344
14141414	55555550	09-DEC-21	1400	56565656

Figure 11: bid table after insertion

- Insert billing data

```
insert into Billing values
  ('11000000', date '2021-12-01','22222222');

insert into Billing values
  ('22000000', date '2021-12-03','44444444');

insert into Billing values
  ('33000000', date '2021-12-04','77777777');

insert into Billing values
  ('44000000', date '2021-12-07','88888888');

insert into Billing values
  ('55000000', date '2021-12-08','12121212');

insert into Billing values
  ('66000000', date '2021-12-09','13131313');
--------------------------------------------------------------------------
```

BILL_ID	BILLING_DATE	BID_ID
11000000	01-DEC-21	22222222
22000000	03-DEC-21	44444444
33000000	04-DEC-21	77777777
44000000	07-DEC-21	88888888
55000000	08-DEC-21	12121212
66000000	09-DEC-21	13131313

Figure 12: billing table after insertion

- Insert feedback data

```
insert into Feedback values
  ('When delivering the item, please put it under my home
door.','33344555','11000000',3);
insert into Feedback values
  ('The bill was paid by Apple pay.','12345678','11000000',3);

insert into Feedback values
  ('When delivering the item, please send to me a message on the
delivery day.','98765432','22000000',5);
insert into Feedback values
  ('The order will be delivered on 20-12-
2021.','01010100','22000000',5);

insert into Feedback values
  ('When delivering the item, please deliver it at
night.','11111110','33000000',2);
insert into Feedback values
  ('Payment will be made in cash upon receipt of the
order.','12654312','33000000',2);


insert into Feedback values
  ('When delivering the item, please deliver it at
morning.','88866555','44000000',5);
```

```
insert into Feedback values
  ('The order will be delivered on 29-12-
2021.','45673883','44000000',5);

insert into Feedback values
  ('When delivering the item, please cover it very well to avoid
breaking.','33344555','55000000',1);
insert into Feedback values
  ('The purchased item is one of the most
requested.','98549854','55000000',1);

insert into Feedback values
  ('When delivering the item, please deliver it any day of the week
except Monday.','55555550','66000000',4);
insert into Feedback values
  ('The bill was paid by Credit card.','34433443','66000000',4);
```

FEEDBACK_DESCRIPTION	CLIENT_ID	BILL_ID	RATING
When delivering the item, please put it under my home door.	33344555	11000000	3
The bill was paid by Credit card.	12345678	11000000	3
When delivering the item, please send to me a message on the delivery day.	98765432	22000000	5
The order will be delivered on 20-12-2021.	01010100	22000000	5
When delivering the item, please deliver it at night.	11111110	33000000	2
Payment will be made in cash upon receipt of the order.	12654312	33000000	2
When delivering the item, please deliver it at morning.	88866555	44000000	5
The order will be delivered on 29-12-2021.	45673883	44000000	5
When delivering the item, please cover it very well to avoid breaking.	33344555	55000000	1
The purchased item is one of the most requested.	98549854	55000000	1
When delivering the item, please deliver it any day of the week except Monday.	55555550	66000000	4
The bill was paid by Apple Pay.	34433443	66000000	4

Figure 13: feedback table after insertion

5.4. View statement and implementation

1-Create a view returning the existing items (not bought) information.

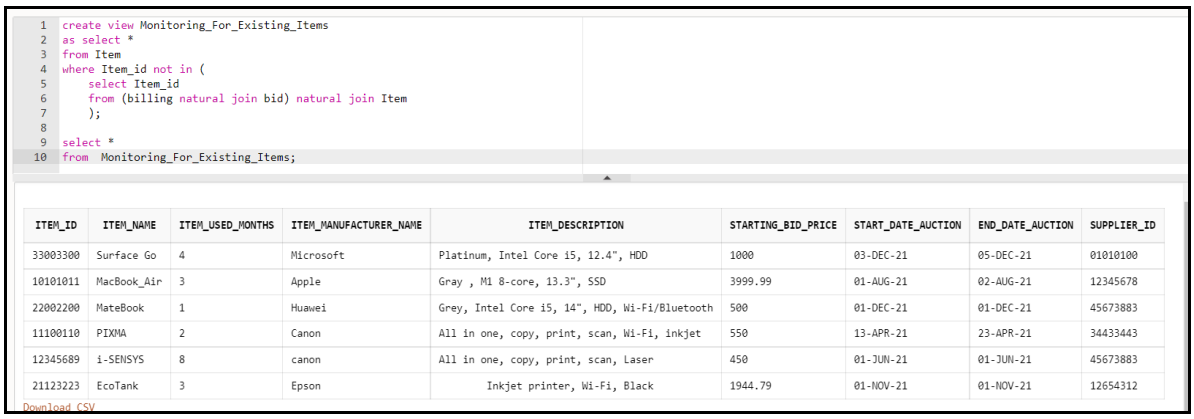
Why it is created: to facilitate the retrieval of available items.

when: it is used any time either by purchaser or supplier.

what is the use of this view: it is used to see the available items.

```
create view Monitoring_For_Existing_Items
as select *
from Item
where Item_id not in (
      select Item_id
      from (billing natural join bid) natural join Item
      );

select *
from  Monitoring_For_Existing_Items;
```



ITEM_ID	ITEM_NAME	ITEM_USED_MONTHS	ITEM_MANUFACTURER_NAME	ITEM_DESCRIPTION	STARTING_BID_PRICE	START_DATE_AUCTION	END_DATE_AUCTION	SUPPLIER_ID
33003300	Surface Go	4	Microsoft	Platinum, Intel Core i5, 12.4", HDD	1000	03-DEC-21	05-DEC-21	01010100
10101011	MacBook_Air	3	Apple	Gray , M1 8-core, 13.3", SSD	3999.99	01-AUG-21	02-AUG-21	12345678
22002200	MateBook	1	Huawei	Grey, Intel Core i5, 14", HDD, Wi-Fi/Bluetooth	500	01-DEC-21	01-DEC-21	45673883
11100110	PIXMA	2	Canon	All in one, copy, print, scan, Wi-Fi, inkjet	550	13-APR-21	23-APR-21	34433443
12345689	i-SENSYS	8	canon	All in one, copy, print, scan, Laser	450	01-JUN-21	01-JUN-21	45673883
21123223	EcoTank	3	Epson	Inkjet printer, Wi-Fi, Black	1944.79	01-NOV-21	01-NOV-21	12654312

Download CSV

Figure 14: first view

2-Create a view returning the slod out (bought) items.

Why it is created: to facilitate the retrieval of not available items.

when: it is used mostly at the end of every week by the suppliers.

what is the use of this view: it is used to see what type of items that bought.

```
create view Sales_information
as select Item_id, Item_name, Bill_id, Proposed_price as
```

```
winner_bid_price, Billing_date
from (billing natural join bid) natural join Item;

select *
from Sales_information
```

```
create view Sales_information
as select Item_id, Item_name, Bill_id, Proposed_price as winner_bid_price, Billing_date
from (billing natural join bid) natural join Item;

select *
from Sales_information
```

ITEM_ID	ITEM_NAME	BILL_ID	WINNER_BID_PRICE	BILLING_DATE
13579135	MacBook_Pro	11000000	8000	01-DEC-21
24680246	HP pavilion	22000000	5500.99	03-DEC-21
09650965	Dell inspiron	33000000	3800	04-DEC-21
99988877	Smart tank	44000000	960.88	07-DEC-21
56565656	EcoTank	55000000	1900	08-DEC-21
11223344	Neverstop	66000000	1200	09-DEC-21

Download CSV

Figure 15: second view

5.5. Queries statement and implementation

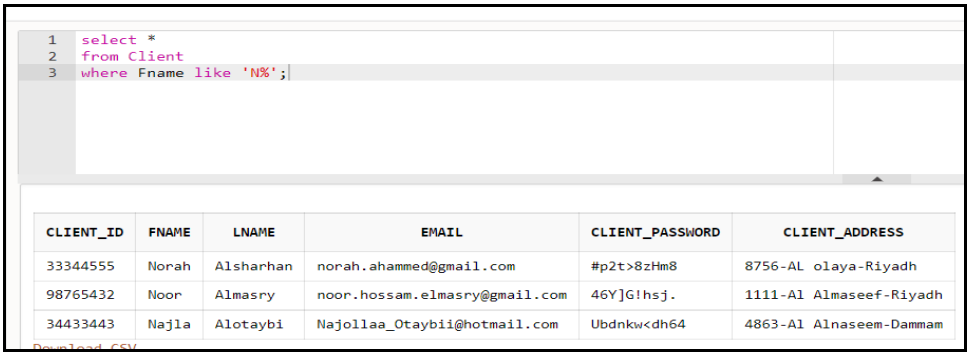
1- Retrieve all clients' details whose first names start with the letter 'N'.

Why it is created: to retrieve specific clients' info.

when: after adding new clients.

what is the use of this query: to classify clients depending on their names first letter.

```
select *
from Client
where Fname like 'N%';
```



```
select *
from Client
where Fname like 'N%';
```

CLIENT_ID	FNAME	LNAME	EMAIL	CLIENT_PASSWORD	CLIENT_ADDRESS
33344555	Norah	Alsharhan	norah.ahammed@gmail.com	#p2t>8zHm8	8756-AL olaya-Riyadh
98765432	Noor	Almasry	noor.hossam.elmasry@gmail.com	46Y]G!hsj.	1111-Al Almaseef-Riyadh
34433443	Najla	Alotaybi	Najollaa_Otaybii@hotmail.com	Ubdnkw<dh64	4863-Al Alnaseem-Dammam

Download CSV

Figure 16: first query

2- Retrieve all printers information that has both 'Arabic' and 'English' languages.

Why it is created: to retrieve printers with a specific feature.

when: there is a purchaser who wants a printer that supports both languages.

what is the use of this query: it is used to classify printers depending on their language.

```
select *
from Printer natural join Item
where Item_id in  ((select Item_id
    from Item_languages
    where Language='Arabic')

    intersect

    (select Item_id
    from Item_languages
    where Language='English'));
```

```
select *
from Printer natural join Item
where Item_id in  ((select Item_id
    from Item_languages
    where Language='Arabic')

    intersect

    (select Item_id
    from Item_languages
    where Language='English'));
```

ITEM_ID	IS_WIRELESS	IS_CONTAIN_SCANNER	INKJET	ITEM_NAME	ITEM_USED_MONTHS	ITEM_MANUFACTURER_NAME	ITEM_DESCRIPTION	STARTING_BID_PRICE	START_DATE_AUCTION	END_DATE_AUCTION	SUPPLIER_ID
99988877	Yes	Yes	Yes	Smart tank	2	HP	All in one, copy, print, scan, Wi-Fi, inkjet	800	20-JUN-21	07-DEC-21	45673883
21123223	Yes	Yes	Yes	EcoTank	3	Epson	Inkjet printer, Wi-Fi, Black	1944.79	01-NOV-21	01-NOV-21	12654312

Figure 17: second query

3-Retrieve items name and items used month for all computers that have RAM equal 16GB.

Why it is created: to retrieve computers with a specific feature.

when: there is a purchaser who wants a computer that has 16GB RAM.

what is the use of this query: it is used to classify computers depending on the specific value of RAM.

```
select Item_name, Item_used_months
from Computer natural join Item
where RAM = '16GB';
```

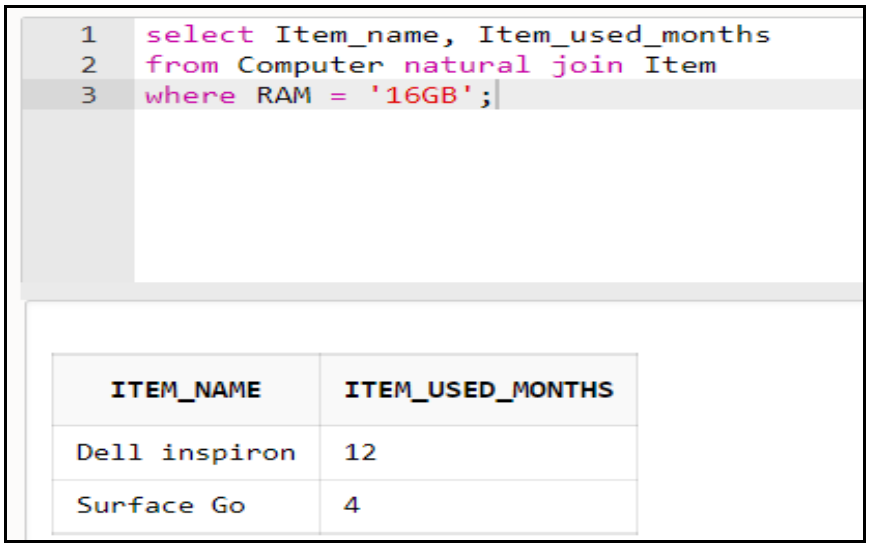


Figure 18: third query

4-For all purchaser who addresses in Riyadh, retrieve their names, IDs, and emails.

Why it is created: to retrieve purchaser with a specific feature.

when: at delivery service.

what is the use of this query: it is used, to sum up, the delivery service for a specific city.

```
select Client_id, Fname, Lname, Email
from Client natural join Purchaser
where Client_address like '%Riyadh%';
```

```
1  select Client_id, Fname, Lname, Email
2  from Client natural join Purchaser
3  where Client_address like '%Riyadh%';
```

CLIENT_ID	FNAME	LNAME	EMAIL
88866555	Raghad	Albosais	raghadkhaled_24@hotmail.com
33344555	Norah	Alsharhan	norah.ahammed@gmail.com
98765432	Noor	Almasry	noor.hossam.elmasry@gmail.com
44444440	Osama	Alsaeed	O_alsaeed@gmail.com

Download CSV

Figure 19: forth query

5-For each item's start date auction, retrieve how many items are in each start date, maximum and minimum start bid price.

Why it is created: to retrieve each item on the first day of the auction.

when: at the end of each first-day auction.

what is the use of this query: to analyze the behavior of auctions and predictions of new items during each auction.

```
select Start_date_auction, count(*) as Items_numbers,
max(Starting_bid_price), min(Starting_bid_price)
from Item
group by Start_date_auction;
```

```
1  select Start_date_auction, count(*) as Items_numbers, max(Starting_bid_price), min(Starting_bid_price)
2  from Item
3  group by Start_date_auction;
```

START_DATE_AUCTION	ITEMS_NUMBERS	MAX(STARTING_BID_PRICE)	MIN(STARTING_BID_PRICE)
01-AUG-21	1	3999.99	3999.99
20-JUN-21	1	800	800
20-SEP-21	1	1000	1000
13-APR-21	1	550	550
02-OCT-21	1	3599.99	3599.99
01-JUN-21	1	450	450
01-NOV-21	1	1944.79	1944.79
20-DEC-20	1	2999.99	2999.99
01-DEC-21	1	500	500
06-JAN-21	1	999.99	999.99
03-DEC-21	1	1000	1000
12-AUG-21	1	6750.99	6750.99

Download CSV

Figure 20: fifth query

6-Retrieve the items id, name, start bid price, the bid id, bid proposed price for the items that have been used less than four months.

Why it is created: to retrieve items that have a specific feature.

when: whenever purchaser wants to make a bid to buy items that have been used less than four months.

what is the use of this query: it is used by the purchaser to see its desired for items with their bids then give large bid price to buy it.

```
select Item_id, Item_name, Starting_bid_price, Bid_id, Proposed_price
from Item natural join Bid
where Item_used_months < 4;
```

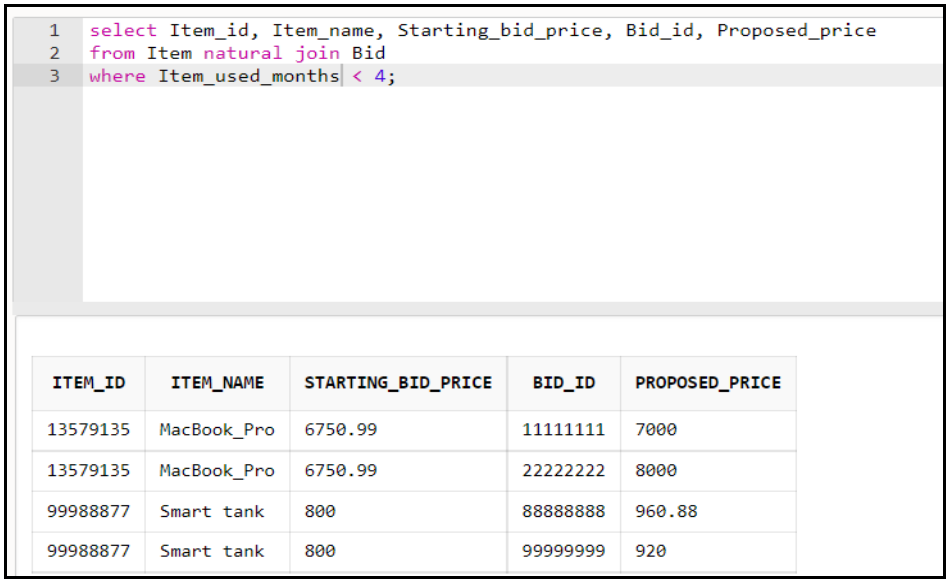


Figure 21: sixth query

7-Retrieve all items id and items name for items that has been bought with its winner bid.

Why it is created: to retrieve all bought items and their winner bids.

when: at any time.

what is the use of this query: suppliers want to see the progress of auctions and the range of bid prices, then it helps them to correctly choose the start bid price. Also will help the purchaser who wants to buy items, to determine which bid price that more likely to be the winner to choose it.

```
select Item_id, Item_name, Proposed_price
where (billing natural join bid) natural join Item;
```

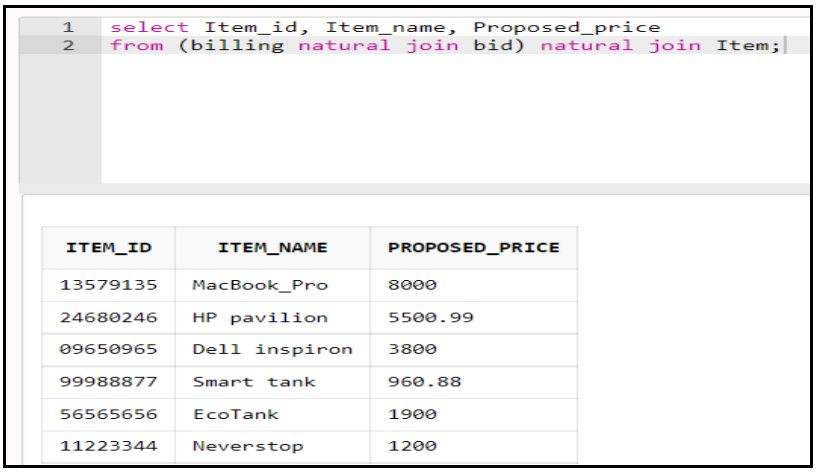


Figure 22: seventh query

8- Retrieve the number of items that have been placed by the supplier whose ID is '33344555'.

Why it is created: to retrieve the items that have been placed by a specific supplier.

when: at any time that supplier wishes to see its items.

what is the use of this query: if one supplier wants to show its own items.

```
select count(*)
from Item
where Supplier_id = '45673883';
```

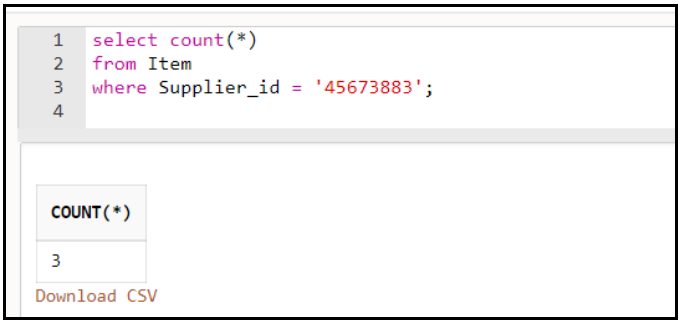


Figure 23: eighth query

9-Update the address to ‘1221-Al Faisaliah-Dammam’ for a client whose ID is '98765432'.

Why it is created: to update the address for a specific client.

when: when the client’s address changes.

what is the use of this query: it is used to update the value of old to a new address.

```
update Client
set Client_address = '1221-Al Faisaliah-Dammam'
where Client_id = '98765432';
```

Before update operation.

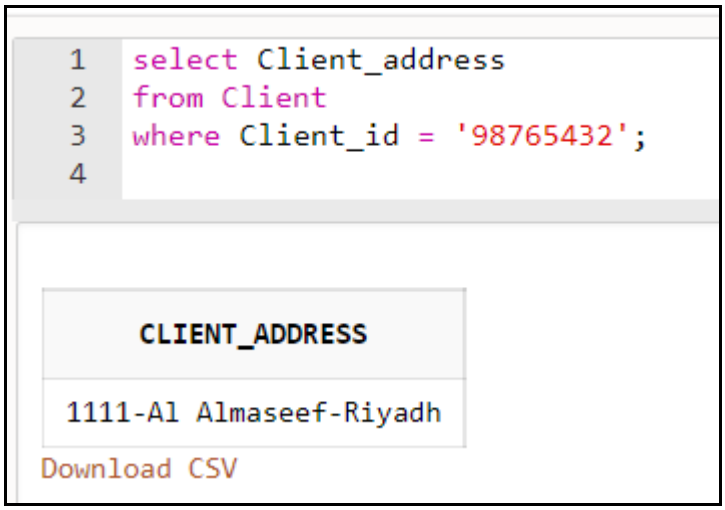


Figure 24: ninth query before update

After update operation.

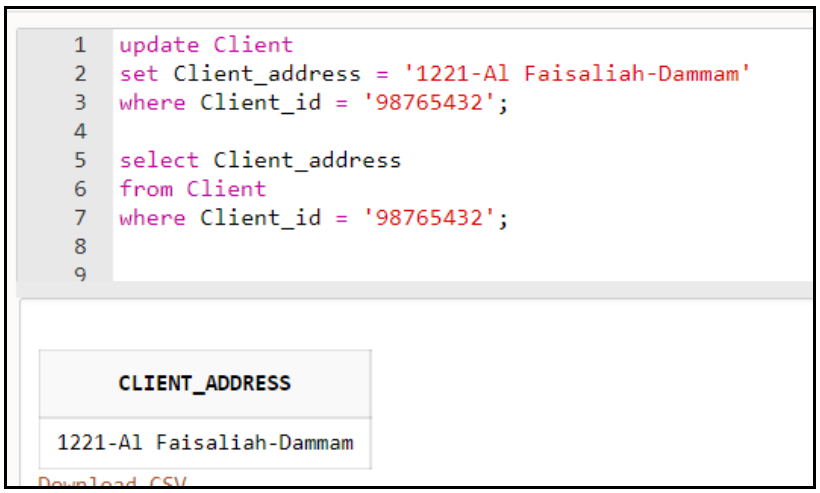


Figure 25: ninth query after update

10-show how many bids are made by the purchaser whose ID is '24680246'.

Why it is created: to retrieve the bids for the specific purchaser.

when: during the auction.

what is the use of this query: it is used to see the purchaser's bids numbers.

```
select count(*)
from Bid
where Client_id ='11111110';
```

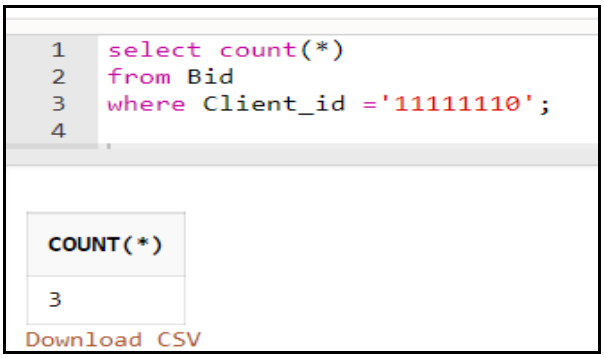


Figure 26: tenth query

11- Retrieve the sum of billings prices for the purchaser whose ID is '88866555'.

Why it is created: to retrieve the total price of purchaser billings.

when: at delivery service.

what is the use of this query: it is used to collect all billings of specific purchaser.

```
select sum(Proposed_price)
from Billing natural join bid
where  Purchaser_id ='33344555';
```

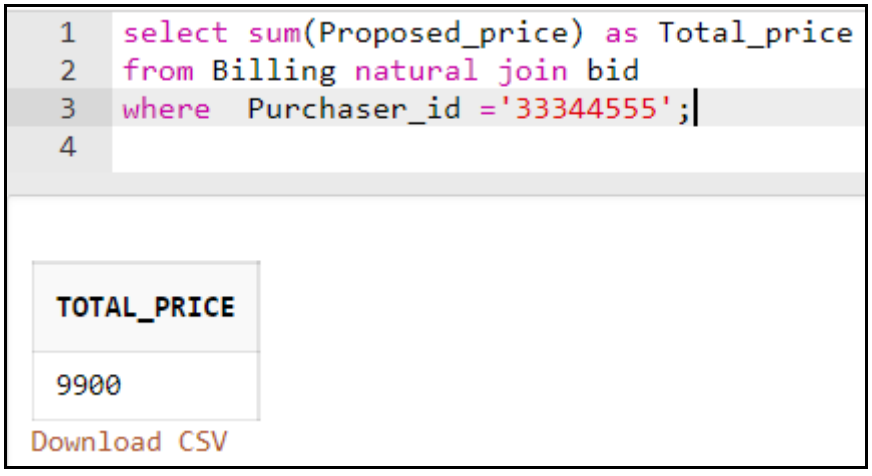


Figure 27: eleventh query

5.6. Triggers

1- Row-Level Trigger

- Create trigger that strips leading and trailing blanks from Fname before they are inserted into Client table.

```
create or replace trigger trim_Fname
before insert on Client
for each row
begin
  :new.Fname := trim(:new.Fname);
end;
```

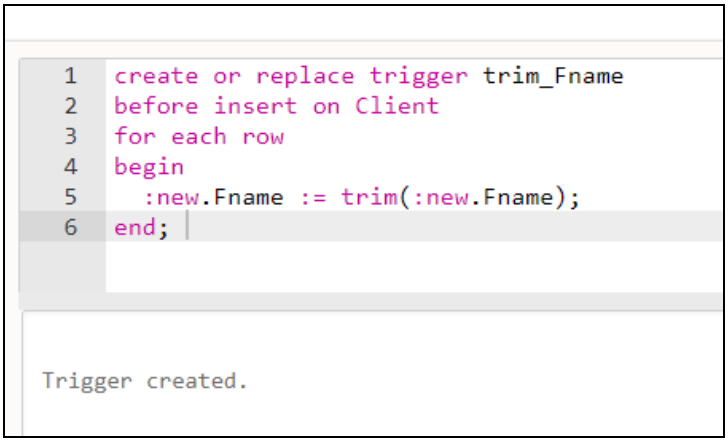


Figure 28: row level trigger (1)

- Insert Fname with leading and trailing blanks.

```
insert into Client values
  ('00000000',    '              Raghad                  ',    'Khaled',
'raghadkhaled_24@hotmail.com', 'Zce*]Y94m+','9999-Al ezdihar-Riyadh' );
```

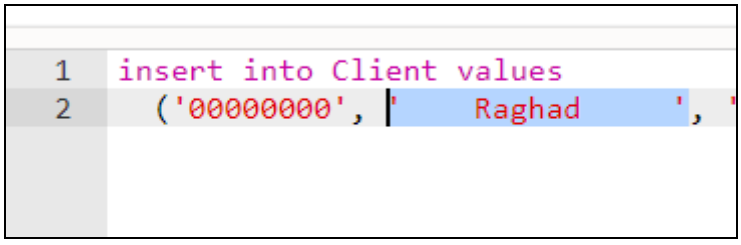


Figure 29: row level trigger (2)

- Display Fname and its lengths.

```
select Fname, length(Fname)
from Client;
```

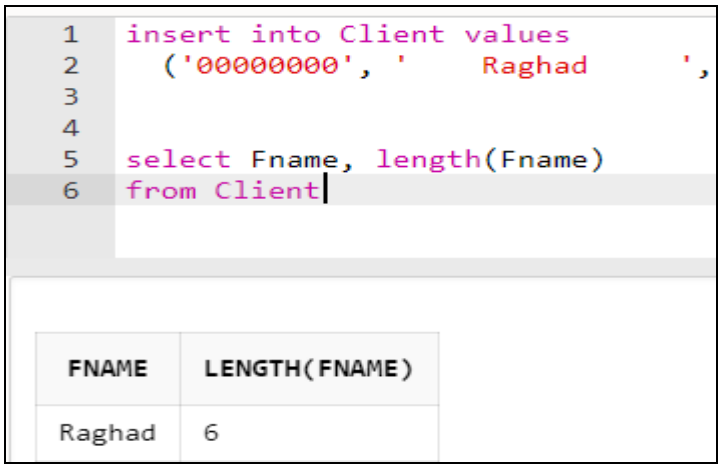


Figure 30: row level trigger (3)

7. Bonus part

We have added three entities, which are Admin, Delivery service, and Mobile.

1- Admin

We added this entity so we can have admins who supervise the buying and selling operations since we can't have an auction without having someone who runs the operation in the right way. We also thought that this entity is important to be included in the DB because there must be someone who supervises the overall operations of the system. This entity has seven attributes (Admin id - Phone number - E-mail – Fname – Lname – Password – Admin address), and it's connected with the client by a relationship (Register) since the admin is the one who signs the clients in the system.

2- Delivery Service

We added this entity to give the purchaser a better experience whit our online auction. This new entity gives us the ability to deliver our items to the one who bought them. We see that this entity is needed since the auction in our case is online and there is no other way for the purchasers to receive their items without delivery service. This entity has three attributes (Order number - Payment method - Shipping address), also this entity has a relationship (Attach) with Billing since we need to include the billing with the delivery.

3- Mobile

We added the entity mobile to be a subclass of the Items and the purpose of doing that was to raise the types of items that can be sold to the purchasers and give them more options to buy. This entity has four attributes (capacity – OS - screen size – color) that describes its features.

7.1. Conceptual EER Model

7.1.1. Assumptions

1-Adimn have only one phone number.

2-There is six admins in this system.

7.1.2. Description of EER data model component

1- Entity and attribute

- Admin is entity and has primary key which is Admin_id, and has set of attributes which are an e-mail, password, client address, phone number, and name is composed attribute that consist of first and last name attributes.
- Mobile entity is subclass for Item entity, has three additional attributes that does not exist in Printer and Computer, which are capacity, OS type, screen size and colour.
- Delivery service is entity and has primary key which is order number, and has set of attributes which are payment method and shipping address.

2-Relationship types and constrains

- Register: each admin should register one or more client, and each client should be registered by only one admin.
- Attach: each delivery service should be attached to one or more billing, and each billing should be attached to only one delivery service.

7.1.3. Conceptual data model

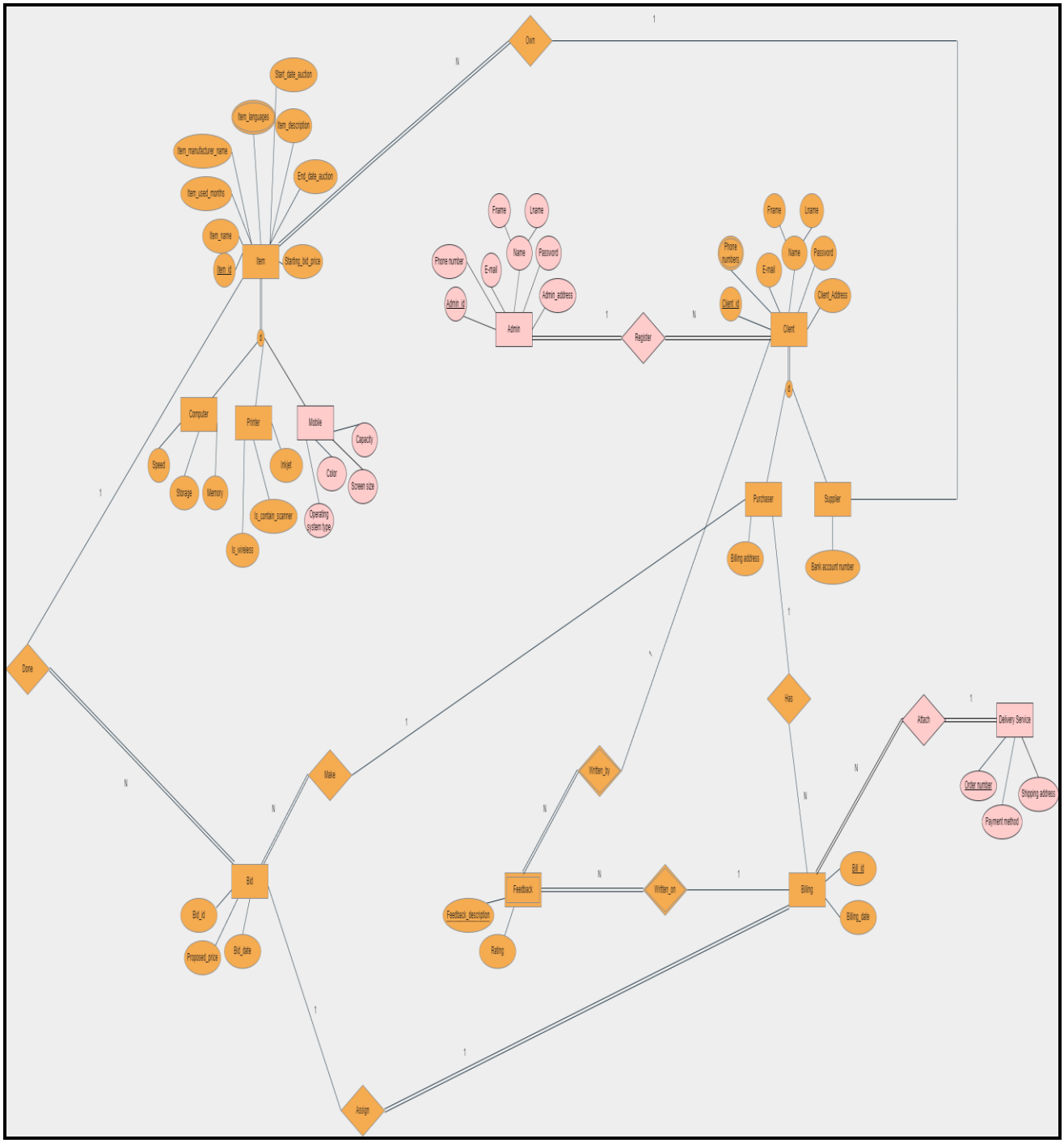


Figure 31: EER diagram with additional entities

7.2. Relational database schema

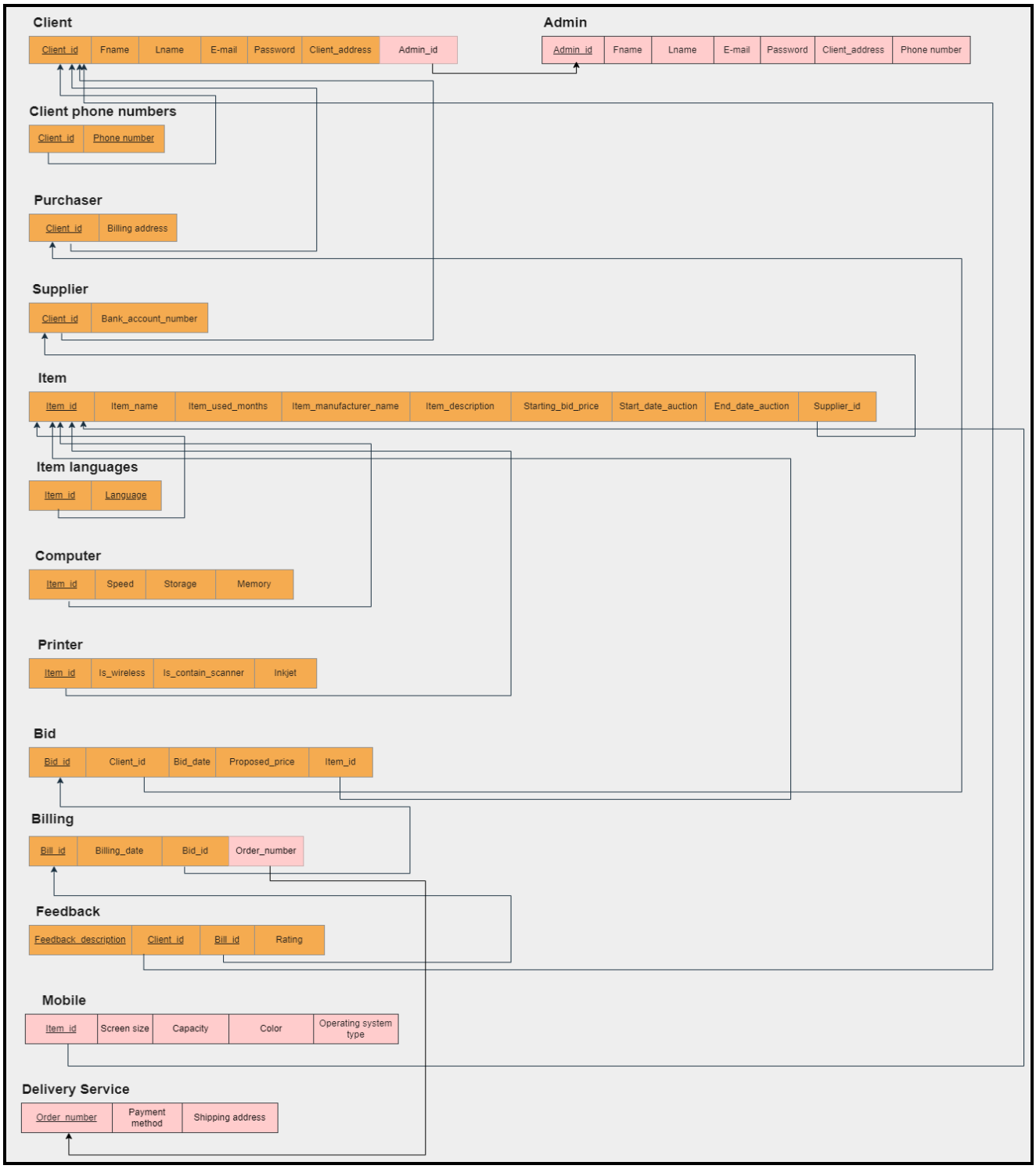


Figure 32: Relational model with additional entities

7.3. Implementation DDL and DML

7.3.1. Define database

- Client table has been added to it one attribute (Admin_id)

```
create table Client (

Client_id  char(8) not null,
Fname    varchar(15) not null,
Lname    varchar(15) not null,
Email   varchar(30) not null,
Client_password varchar(20) not null check
(length(Client_password) >=8 ),
Client_address  varchar(50) not null,
Admin_id  char(8) not null,

primary key (Client_id),
foreign key (Admin_id) references Admin(Admin_id) );
```

- Billing table has been added to it one attribute(Order_number)

```
create table Billing (

Bill_id char(8) not null,
Billing_date date not null,
Bid_id char(8) not null,
Order_number  char(10) not null,

primary key (Bill_id),
foreign key (Bid_id) references Bid(Bid_id),
foreign key (Order_number) references Delivery_Service(Order_number));
```

- Admin table

```
create table Admin (

Admin_id  char(8) not null,
Fname    varchar(15) not null,
Lname    varchar(15) not null,
Email   varchar(30) not null,
Admin_password varchar(20) not null check (length(Admin_password) >=8 ),
Admin_address  varchar(50) not null,
Phone_number char(10) not null,

primary key (Admin_id) );
```

- Mobile table

```
create table Mobile (

Item_id char(8) not null,
Screen_size varchar(15) not null,
Capacity_GB varchar(6) not null,
Colour varchar(10) not null,
Operating_system_type varchar(10) not null,

primary key (Item_id),
foreign key (Item_id) references Item(Item_id) on delete set null);
```

- Delivery service table

```
create table Delivery_Service (

Order_number  char(10) not null,
Payment_method varchar(15),
Shipping_address varchar(50) not null,
primary key (Order_number));
```

7.3.2. Referential integrity constraints

Table	Referencing table(attribute)	Referenced table(attribute)	On DELETE action	On UPDATE action
Mobile	Mobile(Item_id)	Item(Item_id)	SET NULL	NO ACTION

7.3.3. Populate database

- Insert client data, has been added to it value of one attribute(Admin_id)

```
insert into Client values
  ('88866555', 'Raghad', 'Albosais', 'raghadkhaled_24@hotmail.com',
'Zce*]Y94m+','9999-Al ezdihar-Riyadh','75327537' );

insert into Client values
  ('33344555', 'Norah', 'Alsharhan', 'norah.ahammed@gmail.com',
'#p2t>8zHm8', '8756-AL olaya-Riyadh','75327537');

insert into Client values
  ('98765432', 'Noor', 'Almasry', 'noor.hossam.elmasry@gmail.com',
'46Y]G!hsj.', '1111-Al Almaseef-Riyadh','97473702');

insert into Client values
  ('11111110', 'Khaled', 'Alqhatani', 'khaled_alqh@gmail.com',
'}8vDFN<=e;', '0101-Al falah-Dubai','97473702');

insert into Client values
  ('44444440', 'Osama', 'Alsaeed', 'O_alsaeed@gmail.com', 'C85w*$()FN',
'2229-Al malaz-Riyadh','24864905');

insert into Client values
  ('55555550', 'Sara', 'Alshahrani', 'soso_1234@hotmail.com',
'\eny>32Str', '4863-Al nozha-Dahran','24864905');
```

```
insert into Client values
  ('12345678', 'Ali', 'Almutairi', 'Ali_Almutairi@hotmail.com',
'Jeoc@mkf56', '4863-Al Alnaseem-Riyadh','95735027');

insert into Client values
  ('01010100', 'Malak', 'Albaraak', 'baraak_malak@hotmail.com',
'Ihgy\nhkm34', '4863-Al Alnuzha-Dammam','95735027');

insert into Client values
  ('12654312', 'Alaa', 'Alfaise', 'AFaise@hotmail.com', 'Cvbn<hd234',
'4863-Al Alhamraa-Jeddah','97539670');

insert into Client values
  ('45673883', 'hoor', 'Almutairi', 'AlmutairiHOOR@hotmail.com',
'Jfjgin<r4', '4863-Al Alzohour-Dammam','97539670');

insert into Client values
  ('98549854', 'Tariq', 'Alzahrani', 'tariqqqq.zzz@hotmail.com',
'Djirwm@2357', '4863-Al Almurjan-jeddah','79058432');

insert into Client values
  ('34433443', 'Najla', 'Alotaybi', 'Najollaa_Otaybii@hotmail.com',
'Ubdnkw<dh64', '4863-Al Alnaseem-Dammam','79058432');

-----------------------------------------------------------------------
```

CLIENT_ID	FNAME	LNAME	EMAIL	CLIENT_PASSWORD	CLIENT_ADDRESS	ADMIN_ID
88866555	Raghad	Albosais	raghadkhaled_24@hotmail.com	Zce*]Y94m+	9999-Al ezdihar-Riyadh	75327537
33344555	Norah	Alsharhan	norah.ahammed@gmail.com	#p2t>8zHm8	8756-AL olaya-Riyadh	75327537
98765432	Noor	Almasry	noor.hossam.elmasry@gmail.com	46Y]G!hsj.	1111-Al Almaseef-Riyadh	97473702
11111110	Khaled	Alqhatani	khaled_alqh@gmail.com	}8vDFN<=e;	0101-Al falah-Dubai	97473702
44444440	Osama	Alsaeed	O_alsaeed@gmail.com	C85w*$()FN	2229-Al malaz-Riyadh	24864905
55555550	Sara	Alshahrani	soso_1234@hotmail.com	\eny>32Str	4863-Al nozha-Dahran	24864905
12345678	Ali	Almutairi	Ali_Almutairi@hotmail.com	Jeoc@mkf56	4863-Al Alnaseem-Riyadh	95735027
01010100	Malak	Albaraak	baraak_malak@hotmail.com	Ihgy\nhkm34	4863-Al Alnuzha-Dammam	95735027
12654312	Alaa	Alfaise	AFaise@hotmail.com	Cvbn<hd234	4863-Al Alhamraa-Jeddah	97539670
45673883	hoor	Almutairi	AlmutairiHOOR@hotmail.com	Jfjgin<r4	4863-Al Alzohour-Dammam	97539670
98549854	Tariq	Alzahrani	tariqqqq.zzz@hotmail.com	Djirwm@2357	4863-Al Almurjan-jeddah	79058432

Figure 33: client table after adding new column

- Insert billing data, has been added to it value of one attribute(Order_number)

```
insert into Billing values
  ('11000000', date '2021-12-01','22222222', '2356785400');

insert into Billing values
  ('22000000', date '2021-12-03','44444444', '1287540900');

insert into Billing values
  ('33000000', date '2021-12-04','77777777', '4587320900');

insert into Billing values
  ('44000000', date '2021-12-07','88888888', '1156840200');

insert into Billing values
  ('55000000', date '2021-12-08','12121212', '9065389100');

insert into Billing values
  ('66000000', date '2021-12-09','13131313', '1287094500');

-----------------------------------------------------------------------
```

BILL_ID	BILLING_DATE	BID_ID	ORDER_NUMBER
11000000	01-DEC-21	22222222	2356785400
22000000	03-DEC-21	44444444	1287540900
33000000	04-DEC-21	77777777	4587320900
44000000	07-DEC-21	88888888	1156840200
55000000	08-DEC-21	12121212	9065389100
66000000	09-DEC-21	13131313	1287094500

Figure 34: billing table after adding new column

- Insert admin data

```
insert into Admin values
('75327537','Ahmed','Alshehri','Ahmed222@gmail.com','asd##993','1231-Al
shobra-Riyadh','0567893279');

insert into Admin values
('97473702','Ali','Alotaibi','Ali99@gmail.com','lpyt@45h','8965-Al
azizia-Riyadh','0555438690');



insert into Admin values
('24864905','Rawan','Alshlawi','Rawan66@gmail.com','lmitf#*1','8747-Al
yarmouk-Riyadh','0550382017');

insert into Admin values
('95735027','Turki','Algheith','Turki-3@gmail.com','tnpre@@90z','1179-
Al nmar-Riyadh','0550543961');

insert into Admin values
('97539670','Danah','Alshikh','Danah76@gmail.com','hyremouw6','8054-Al
hamra-Riyadh','0567984367');

insert into Admin values
('79058432','Salma','Almubarak','salma456@gmail.com','pouy@#98u','3487-
Al mughrazat-Riyadh','0555209874');
```

ADMIN_ID	FNAME	LNAME	EMAIL	ADMIN_PASSWORD	ADMIN_ADDRESS	PHONE_NUMBER
75327537	Ahmed	Alshehri	Ahmed222@gmail.com	asd##993	1231-Al shobra-Riyadh	0567893279
97473702	Ali	Alotaibi	Ali99@gmail.com	lpyt@45h	8965-Al azizia-Riyadh	0555438690
24864905	Rawan	Alshlawi	Rawan66@gmail.com	lmitf#*1	8747-Al yarmouk-Riyadh	0550382017
95735027	Turki	Algheith	Turki-3@gmail.com	tnpre@@90z	1179-Al nmar-Riyadh	0550543961
97539670	Danah	Alshikh	Danah76@gmail.com	hyremouw6	8054-Al hamra-Riyadh	0567984367
79058432	Salma	Almubarak	salma456@gmail.com	pouy@#98u	3487-Al mughrazat-Riyadh	0555209874

Figure 35: admin table after insertion

- Insert mobile data, also we insert mobiles information in Item table

-data at Item table

```
insert into Item values
('98765034','iphone13 mini',12,'Apple','white ,128GB,5.4", 5G ,ios
15',2500.00, date '2021-01-01', date '2021-01-10','12345678');

insert into Item values
('24587569','Huawei Nova 9',10,'Huawei','Black ,128GB ,6.57", 4G LTE,
Android 11',900.00, date '2021-02-01', date '2021-02-10','01010100');

insert into Item values
('11223759','Huawei Y9a',8,'Huawei','Silver ,128GB ,6.63", 4G LTE,
Android 10',770.00, date '2021-03-01', date '2021-03-10','12654312');

insert into Item values
('44887290','Samsung Galaxy',9,'Samsung','Red ,128GB,6.5", 5G ,Android
10',990.00, date '2021-04-01', date '2021-04-10','45673883');

insert into Item values
('12740097','Samsung Galaxy',7,'Samsung','Black ,256GB,6.7",
5G ,Android 11',2700.00, date '2021-05-01', date '2021-05-
10','98549854');

insert into Item values
```

```
('87540932','iphone11 ',24,'Apple','Green ,128GB,6.1", 4G LTE ,ios
13',1330.00, date '2021-06-01', date '2021-06-10','34433443');
-----------------------------------------------------------------------
```

98765034	iphone13 mini	12	Apple	white ,128GB,5.4", 5G ,ios 15	2500	01-JAN-21	10-JAN-21	12345678
24587569	Huawei Nova 9	10	Huawei	Black ,128GB ,6.57", 4G LTE, Android 11	900	01-FEB-21	10-FEB-21	01010100
11223759	Huawei Y9a	8	Huawei	Silver ,128GB ,6.63", 4G LTE, Android 10	770	01-MAR-21	10-MAR-21	12654312
87540932	iphone11	24	Apple	Green ,128GB,6.1", 4G LTE ,ios 13	1330	01-JUN-21	10-JUN-21	34433443
44887290	Samsung Galaxy	9	Samsung	Red ,128GB,6.5", 5G ,Android 10	990	01-APR-21	10-APR-21	45673883
12740097	Samsung Galaxy	7	Samsung	Black ,256GB,6.7", 5G ,Android 11	2700	01-MAY-21	10-MAY-21	98549854

Figure 36: Item table after inserting mobile data

-data at Mobile table

```
insert into Mobile values
('98765034','display 5.4"','128GB' ,'White' ,'ios 15');

insert into Mobile values
('24587569','display 6.57"','128GB' ,'Black' ,'Android 11');

insert into Mobile values
('11223759','display 6.63"','128GB' ,'Silver' ,'Android 10');

insert into Mobile values
('44887290','display  6.5"','128GB' ,'Red' ,'Android 10');


insert into Mobile values
('12740097','display 6.7"','256GB' ,'Black' ,'Android 11');

insert into Mobile values
('87540932','display 6.1"','128GB' ,'Green' ,'ios 13');
-----------------------------------------------------------------------
```

ITEM_ID	SCREEN_SIZE	CAPACITY_GB	COLOUR	OPERATING_SYSTEM_TYPE
98765034	display 5.4"	128GB	White	ios 15
24587569	display 6.57"	128GB	Black	Android 11
11223759	display 6.63"	128GB	Silver	Android 10
44887290	display 6.5"	128GB	Red	Android 10
12740097	display 6.7"	256GB	Black	Android 11
87540932	display 6.1"	128GB	Green	ios 13

Figure 37: mobile table after insertion

- Insert delivery service data

```
insert into Delivery_Service values
('2356785400','Apple pay','7760-AL yasmin-Riyadh');

insert into Delivery_Service values
('1287540900','Cash','6520-AL ghadir-Riyadh');

insert into Delivery_Service values
('4587320900','Cash','5560-AL waha-Riyadh');

insert into Delivery_Service values
('1156840200','Credit card','8380-AL olaya-Riyadh');

insert into Delivery_Service values
('9065389100','Credit card','7760-AL yasmin-Riyadh');

insert into Delivery_Service values
('1287094500','Credit card','4570-AL narjes-Riyadh');

-------------------------------------------------------------------------
```

ORDER_NUMBER	PAYMENT_METHOD	SHIPPING_ADDRESS
2356785400	Apple pay	7760-AL yasmin-Riyadh
1287540900	Cash	6520-AL ghadir-Riyadh
4587320900	Cash	5560-AL waha-Riyadh
1156840200	Credit card	8380-AL olaya-Riyadh
9065389100	Credit card	7760-AL yasmin-Riyadh
1287094500	Credit card	4570-AL narjes-Riyadh

Figure 38: delivery service table after insertion

7.3.4. Views and implementation in SQL.

1-Create a view of delivery orders to be delivered with its billings information.

why it is created: to facilitate to retrieve all billing that needs to be delivered.

when: every day.

what is the use of this view: to show the delivery orders then classify the addresses to be delivered.

```
create view Delivery_order
as select *
from Delivery_Service natural join Billing;

select *
from Delivery_order;
```

```
1  create view Delivery_order
2  as select *
3  from Delivery_Service natural join Billing;
4
5  select *
6  from Delivery_order;
7
8
```

ORDER_NUMBER	PAYMENT_METHOD	SHIPPING_ADDRESS	BILL_ID	BILLING_DATE	BID_ID
2356785400	Apple pay	7760-AL yasmin-Riyadh	11000000	01-DEC-21	22222222
1287540900	Cash	6520-AL ghadir-Riyadh	22000000	03-DEC-21	44444444
4587320900	Cash	5560-AL waha-Riyadh	33000000	04-DEC-21	77777777
1156840200	Credit card	8380-AL olaya-Riyadh	44000000	07-DEC-21	88888888
9065389100	Credit card	7760-AL yasmin-Riyadh	55000000	08-DEC-21	12121212
1287094500	Credit card	4570-AL narjes-Riyadh	66000000	09-DEC-21	13131313

Download CSV

Figure 39: first view in BONUS part

7.3.5. Quires and implementation in SQL.

1-Which delivery orders have shippling adddress equal to '7760-AL yasmin-Riyadh'.

Why it is created: to retrieve all orders for specific purchasers.

when: at delivery service.

what is the use of this query: to collect all items for one purchaser then deliver them together.

```
select *
from Delivery_Service
where  Shipping_address ='7760-AL yasmin-Riyadh';
```

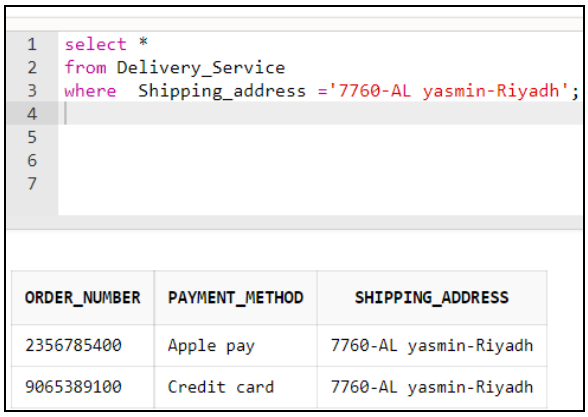


Figure 40: first query in BONUS part

2- Retrieve each client name and ID with its admin ID.

Why it is created: to retrieve clients with its admins.

when: any time client wants to know about its admin.

what is the use of this query: if the client needs to help, then see who its admin to contact with it.

```
select Client_id, Fname as Client_Fname, Admin_id
from Client;
```

```
1  select Client_id, Fname as Client_Fname, Admin_id
2  from Client;
3
```

CLIENT_ID	CLIENT_FNAME	ADMIN_ID
88866555	Raghad	75327537
33344555	Norah	75327537
98765432	Noor	97473702
11111110	Khaled	97473702
44444440	Osama	24864905
55555550	Sara	24864905
12345678	Ali	95735027
01010100	Malak	95735027
12654312	Alaa	97539670
45673883	hoor	97539670
98549854	Tariq	79058432

Figure 41: second query in BONUS part