CC5051 Databases
Coursework Assignment 2022/23 Autumn
The Mobile for You Case Study

Question 1 - Initial ER Model

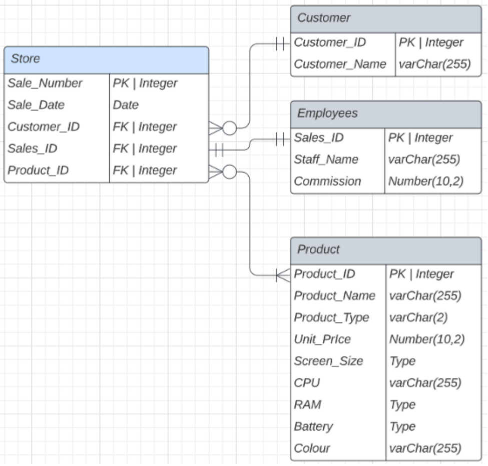
Draw an initial Entity Relationship model, stating any assumptions you make

- Products: This entity would include all the products sold by 'Mobile for You', including smart phones, tablets, and accessories. The attributes of this entity might include OS, processor speed, memory capacity, screen size, camera, power, colour, and price.
- Sales: The attributes of this entity would include the sale number of the items that were sold, customers name, the date of the sale and the total amount of the sale
- Staff: This entity would include information about the sales staff and managers at Mobile for You. The attributes of this entity might include the staff member's name, position, and commission earned.

Each product can be sold multiple times and therefore there are many sales for each product in a (1:M) relationship. Each salesperson can make multiple sales (1:M).

Some assumptions that might be made in this model include:

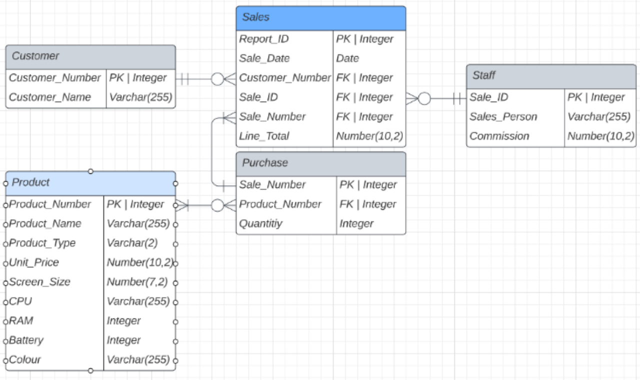
- Each product has only one set of attributes, such as OS, processor speed, etc.
- Each sale is for either one or many products product.
- Each salesperson has only one position (e.g., salesperson or manager).
- Each sale can only be handled by one employee (otherwise the commission can't be calculated correctly).


Store
Sale_Number | PK | Integer
Sale_Date | Date
Customer_ID | FK | Integer
Sales_ID | FK | Integer
Product_ID | FK | Integer
Customer
Customer_ID | PK | Integer
Customer_Name | varChar(255)
Employees
Sales_ID | PK | Integer
Staff_Name | varChar(255)
Commission | Number(10,2)
Product
Product_ID | PK | Integer
Product_Name | varChar(255)
Product_Type | varChar(2)
Unit_PrIce | Number(10,2)
Screen_Size | Type
CPU | varChar(255)
RAM | Type
Battery | Type
Colour | varChar(255)

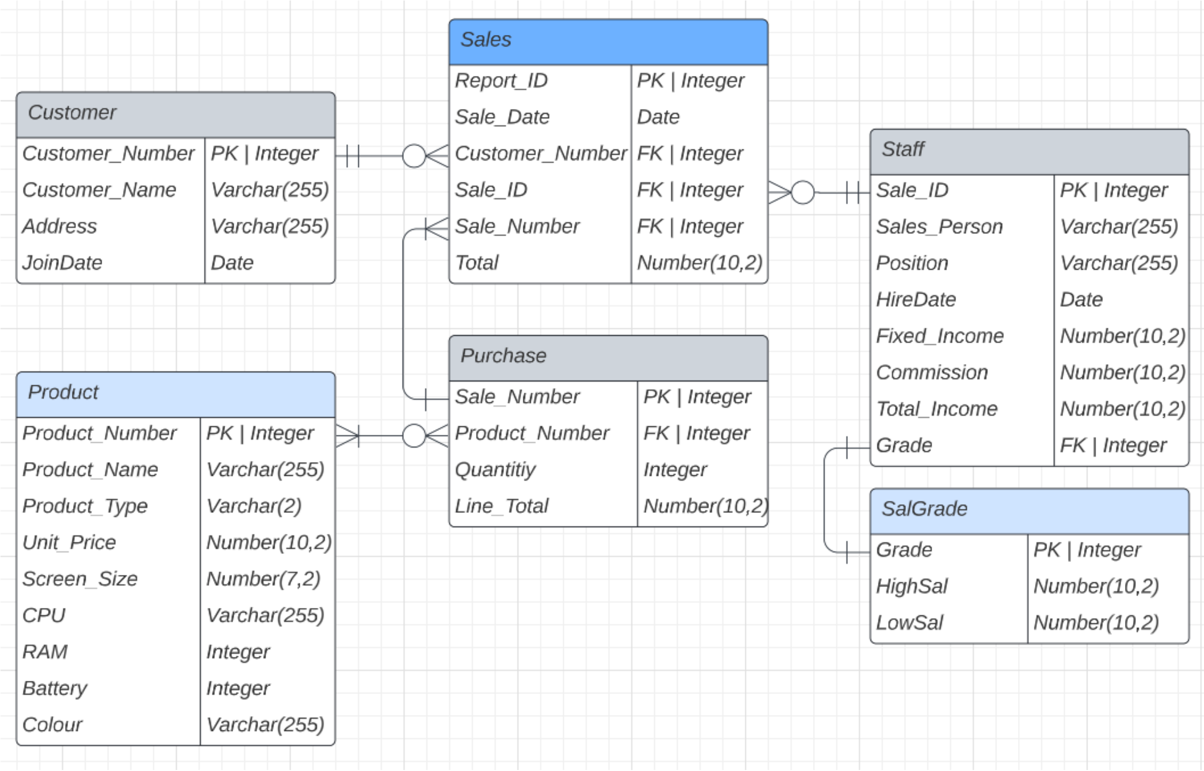
Question 2 - Normalisation

Normalise (to third normal form) the Sales Form. Show all the steps of normalisation clearly. State any assumptions. Now draw an Entity-Relationship diagram for these normalised entities.

The normalisation process includes three steps: removing repeating groups, removing partial dependencies, and removing transitive dependencies. When making purchases the data would be repeating. For example, customers can buy one or more products if they would like to and if they chose to buy more than one product then the information would be repeating in the initial ER diagram for every new product that the customer would have bought. This led me to splitting the store table into two new tables called Purchase, and Sales. This would allow the Sales Number in the Purchase table to be comprise many Sales Numbers in the Sales Table in a many to one form. The sales Table can contain many purchases, but purchases can only be made in one sale.


Sales
Report_ID | PK | Integer
Sale_Date | Date
Customer_Number | FK | Integer
Sale_ID | FK | Integer
Sale_Number | FK | Integer
Line_Total | Number(10,2)
Customer
Customer_Number | PK | Integer
Customer_Name | Varchar(255)
Staff
Sale_ID | PK | Integer
Sales_Person | Varchar(255)
Commission | Number(10,2)
Purchase
Sale_Number | PK | Integer
Product_Number | FK | Integer
Quantitiy | Integer
Product
Product_Number | PK | Integer
Product_Name | Varchar(255)
Product_Type | Varchar(2)
Unit_Price | Number(10,2)
Screen_Size | Number(7,2)
CPU | Varchar(255)
RAM | Integer
Battery | Integer
Colour | Varchar(255)

I also came up with more information that could be used for the employees such as salary and position. Sale date is on the sales table rather than the purchase table otherwise the date will be repeating each time when they were all made on the same day at the same time and would just be easier to see on the main table. This is a clean bridge for the manager to access the main information for the day which would be how much was made per purchase, who made the most sales and the customer who it was sold to. The final ER diagram looks like:



Sales	
Report_ID	PK | Integer
Sale_Date	Date
Customer_Number	FK | Integer
Sale_ID	FK | Integer
Sale_Number	FK | Integer
Total	Number(10,2)

Customer	
Customer_Number	PK | Integer
Customer_Name	Varchar(255)
Address	Varchar(255)
JoinDate	Date

Staff	
Sale_ID	PK | Integer
Sales_Person	Varchar(255)
Position	Varchar(255)
HireDate	Date
Fixed_Income	Number(10,2)
Commission	Number(10,2)
Total_Income	Number(10,2)
Grade	FK | Integer

Purchase	
Sale_Number	PK | Integer
Product_Number	FK | Integer
Quantitiy	Integer
Line_Total	Number(10,2)

Product	
Product_Number	PK | Integer
Product_Name	Varchar(255)
Product_Type	Varchar(2)
Unit_Price	Number(10,2)
Screen_Size	Number(7,2)
CPU	Varchar(255)
RAM	Integer
Battery	Integer
Colour	Varchar(255)

SalGrade	
Grade	PK | Integer
HighSal	Number(10,2)
LowSal	Number(10,2)

Question 3 - Table Creation

<u>Give the relational schema (set of tables) that you will create.</u>

```
create table product(/*Product info, i.e CPU speeds, colour etc.*/
Product_Number integer,
Product_Name varchar(255),
Product_Type varchar(2),
Unit_Price number(10,2),
Screen_Size number(10,2),
CPU varchar(255),
RAM integer,
Battery integer,
Colour varchar(255),
Primary Key(Product_number)
);

create table customer(/*Customer information*/
Customer_Number integer,
Customer_Name varchar(255),
Address varchar(255),
JoinDate Date,/*date customer made first purchase /when account was made*/
Primary Key (Customer_number)
);
```

```
Create Table SalGrade(
Grade Integer,
Lowsal number(10,2),
HighSal number(10,2),
Primary Key(Grade)
);

Create table Staff(/* Employee information table */
Sale_ID integer,
Sales_Person varchar(255),
Position varchar(255),
Hiredate Date,
FixedIncome number(10,2),/*fixed monthly pay before commission is added*/
Commission number(10,2),
TotalIncome number(10,2),/*total monthly pay after commission is added*/
Grade integer,
Primary key(Sale_ID),
Foreign Key(Grade) references SalGrade(Grade)
);

Create table Purchase(/*Info on the sales i.e how many products were sold*/
Sale_Number integer,
Product_Number integer,
Quantity integer,
Line_Total number(10,2),
Primary KEY (Sale_number),
foreign key (product_number) references product(product_number)
);

create table sales(/*Main sales table*/
Report_ID integer,
Sale_Date date,
customer_number integer,
Sale_ID integer,
Sale_Number integer,
Total number(10,2),
Primary Key (Report_ID),
foreign key (customer_number) references customer(customer_number),
foreign key (sale_id) references staff(sale_id),
foreign key (Sale_number) references purchase(sale_number)
);
```

Question 4 – Implementation

Implement and submit printed documentation for the following.

Part A – Testing

(a) Create the tables you have identified for the 'Mobile for You' database and populate them with appropriate test data of your choice.

```
/*--------Product Table------------*/
insert into product
values(1,'Iphone 14', 'SP', 849, 6.1, 'A15 Bionic chip', 6, 4323, 'Red');
insert into product
values(2,'Iphone 14 plus', 'SP', 1099, 6.7, 'A15 Bionic chip', 6, 4323,
'Black');
insert into product
values(3,'Ipad Pro 11inch', 'TB', 839, 11, 'Apple M2 chip', 16, 7538,
'Blue');
```

```
insert into product
values(4,'Ipad Air 5th gen', 'TB', 669, 10.9, 'Apple M1 chip', 8, 8827,
'Space Grey');
insert into product(product_number, product_name, product_type, unit_price,
colour)
values(5,'Iphone USB Cable', 'AC', 20,'White');
/*--------Customer Table------------*/
insert into customer
values (1, 'Albert Einstein', '8 Manchester Road',
to_date('17-JUL-2022','DD-MON-YYYY'));
insert into customer
values (2, 'Tyler the Creator', '36 Chester Road',
to_date('28-JUL-2022','DD-MON-YYYY'));
insert into customer
values (3, 'Travis Scott', '71 School Lane',
to_date('03-SEP-2022','DD-MON-YYYY'));
insert into customer
values (4, 'Elon Musk', '30 Church Street',
to_date('07-OCT-2022','DD-MON-YYYY'));
insert into customer
values (5, 'Kanye West', '64 South Street',
to_date('17-OCT-2022','DD-MON-YYYY'));
/*--------Salary Grade Table------------*/
insert into SalGrade
values (1,  700, 1200);

insert into SalGrade
values (2, 1201, 1400);

insert into SalGrade
values (3, 1401, 2000);

insert into SalGrade
values (4, 2001, 3000);

insert into SalGrade
 values (5, 3001, 9999);
/*--------Staff Table------------*/
insert into staff
values (1, 'Oliver','Manager', to_date('20-JUN-2022','DD-MON-YYYY'),
4166, 1700.2, 5866.2, 5);

insert into staff
values (2, 'Jess','Manager', to_date('21-JUN-2022','DD-MON-YYYY'),
3200, 1482.21, 4682.21, 5);

insert into staff
values (3,'Emma','Sales Associate', to_date('04-JUL-2022','DD-MON-YYYY'),
1780, 807.7, 2587.7, 4);

insert into staff
values (4,'Liam','Sales Associate', to_date('26-AUG-2022','DD-MON-YYYY'),
950, 350.2, 1300.20, 2);

insert into staff
values (5, 'Noah','Sales Associate', to_date('01-DEC-2022','DD-MON-YYYY'),
1899, 0, 1899, 3);
/*--------Purchase Table------------*/
insert into purchase
values(1, 2, 2, 2198);
```

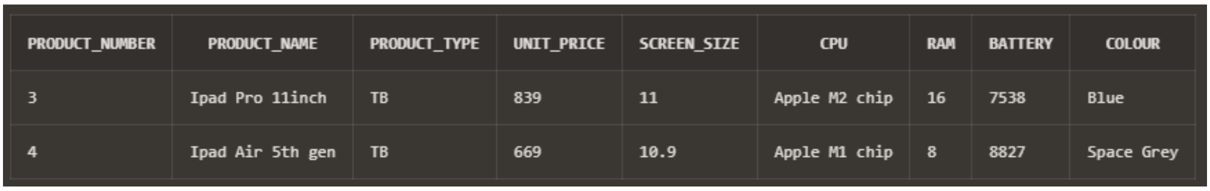
```
insert into purchase
values(2, 1, 2, 1698);
insert into purchase
values(3, 4, 3, 2007);
insert into purchase
values(4, 3, 1,839);
insert into purchase
values(5, 4, 1, 669);
/*--------Sales Table------------*/
insert into sales
values (1, to_date('17-JUL-2022','DD-MON-YYYY'), 1, 2, 1, 2198);
insert into sales
values (2, to_date('28-JUL-2022','DD-MON-YYYY'), 2, 3, 2, 1698);
insert into sales
values (3, to_date('02-SEP-2022','DD-MON-YYYY'), 1, 4, 3, 2007);
insert into sales
values (4, to_date('03-SEP-2022','DD-MON-YYYY'), 3, 4, 4, 837);
insert into sales
values (5, to_date('07-OCT-2022','DD-MON-YYYY'), 4, 4, 5, 669);
```

Part B - Implementation

(b) Implement all the product information queries and all the sales process queries. You should provide, for each query, a separate Oracle script and the resulting output. Number the queries Q1, Q2, Q3, Q4, Q5, Q6.

Q1. what product models have a power capacity of at least 5000 mAh.

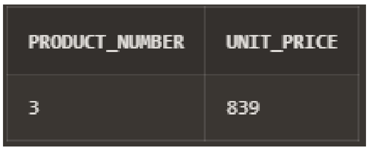
```
select * from product where battery > 5000;
```



PRODUCT_NUMBER	PRODUCT_NAME	PRODUCT_TYPE	UNIT_PRICE	SCREEN_SIZE	CPU	RAM	BATTERY	COLOUR
3	Ipad Pro 11inch	TB	839	11	Apple M2 chip	16	7538	Blue
4	Ipad Air 5th gen	TB	669	10.9	Apple M1 chip	8	8827	Space Grey

Q2. find the model number and price of all products (of any type) in 'Blue' colour.

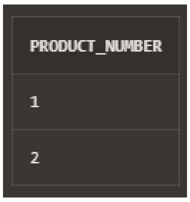
```
select product_number, unit_price from product where colour = 'Blue';
```



PRODUCT_NUMBER	UNIT_PRICE
3	839

Q3. find the model numbers of all smart phone with more than 6GB memory.

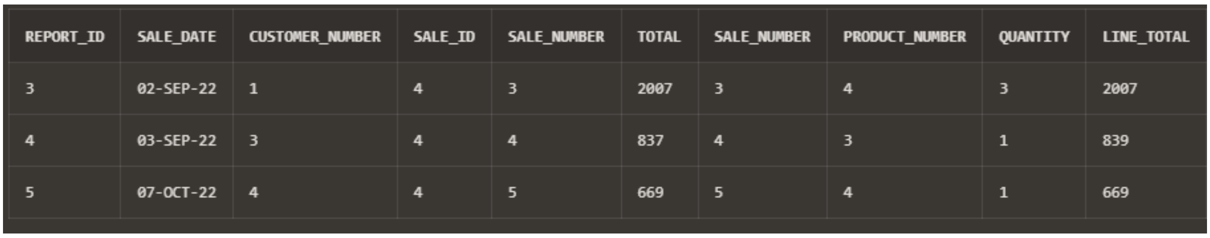
```
select product_number from product

where RAM >= 6 and product_type = 'SP';
```



PRODUCT_NUMBER
1
2

Q4. details of sales that are handled by an individual salesperson.

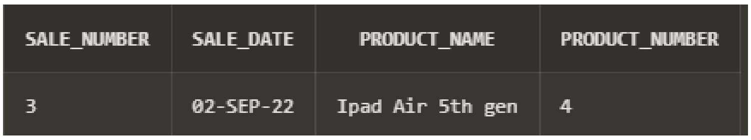
```
select * from sales
join purchase
on sales.sale_number = purchase.sale_number
where sale_id = 4;
```



REPORT_ID	SALE_DATE	CUSTOMER_NUMBER	SALE_ID	SALE_NUMBER	TOTAL	SALE_NUMBER	PRODUCT_NUMBER	QUANTITY	LINE_TOTAL
3	02-SEP-22	1	4	3	2007	3	4	3	2007
4	03-SEP-22	3	4	4	837	4	3	1	839
5	07-OCT-22	4	4	5	669	5	4	1	669

Q5. list of tablets which have been sold on a particular day.

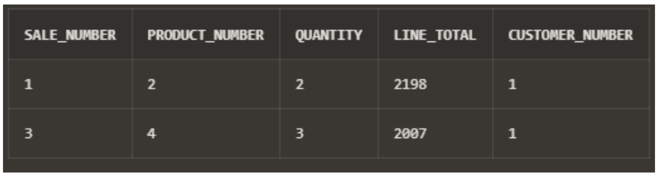
```
select purchase.sale_number, sale_date, product.product_name, product.product_number
from sales
join purchase
on sales.sale_number = purchase.sale_number
join product
on purchase.product_number = product.product_number
where product_type = 'TB'
 and sale_date = '02-SEP-2022';
```



SALE_NUMBER	SALE_DATE	PRODUCT_NAME	PRODUCT_NUMBER
3	02-SEP-22	Ipad Air 5th gen	4

Q6. details of the purchases a particular customer has made.

```
select purchase.*, sales.customer_number
from purchase
join sales
on sales.sale_number = purchase.sale_number
where customer_number = 1;
```



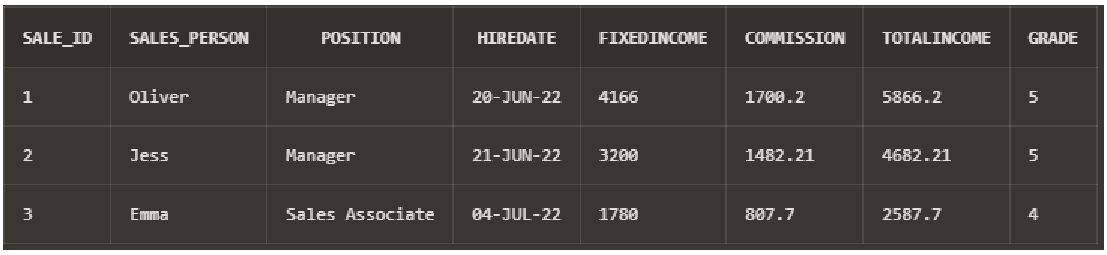
SALE_NUMBER	PRODUCT_NUMBER	QUANTITY	LINE_TOTAL	CUSTOMER_NUMBER
1	2	2	2198	1
3	4	3	2007	1

Part C – Custom Testing

(c) Now produce two additional queries (Q7 and Q8) which you believe will be useful to the company. Consider extending your schema with either extra data columns or tables for this purpose. Give a clear textual description of what each of your two queries is supposed to show. Use your judgement as to what information the queries should contain and how they should be formatted.

Q7. Custom Query - Details of staff members who have earned more than $800 in commission and have a salary grade of 4 or higher

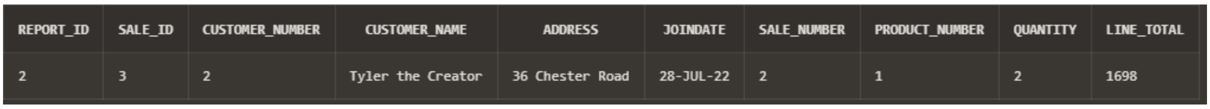
```
select *
from staff
where commission > 800
and grade > 3;
```



SALE_ID	SALES_PERSON	POSITION	HIREDATE	FIXEDINCOME	COMMISSION	TOTALINCOME	GRADE
1	Oliver	Manager	20-JUN-22	4166	1700.2	5866.2	5
2	Jess	Manager	21-JUN-22	3200	1482.21	4682.21	5
3	Emma	Sales Associate	04-JUL-22	1780	807.7	2587.7	4

Q8. Custom Query - Sales report with details of customers and their purchases where number of products bought was 2 or more and total price was less than 2000

```
select sales.report_id, sales.sale_id, customer.*,purchase.*
from sales
join customer
on customer.customer_number=sales.customer_number
join purchase
on sales.sale_number=purchase.sale_number
where quantity > 1
and line_total < 2000;
```



REPORT_ID	SALE_ID	CUSTOMER_NUMBER	CUSTOMER_NAME	ADDRESS	JOINDATE	SALE_NUMBER	PRODUCT_NUMBER	QUANTITY	LINE_TOTAL
2	3	2	Tyler the Creator	36 Chester Road	28-JUL-22	2	1	2	1698

Summarisation

The ER diagram and rational schema data that was created clearly represent what kind of data will be used and recorded while the store is open. I believe that the tables created will be very useful to Dave as it will simplify the information that he is dealing with and make it easy to access and access any information.