PATIENTS

Patient (patientNo, patName, patAddress, DOB)
Ward (wardNo, wardName, wardType, noOfBeds)
Contains (patientNo, wardNo, admissionDate)
Drug (drugNo, drugName, costPerUnit)
Prescribed (patientNo, drugNo, unitPersday, startDate, finishDate)

1. List all patients details, alphabetically
   SELECT * FROM PATIENT
   ORDERBY PATIENTNAME

2. List the patient number and the patient name of all patients currently in the Surgical(wardName)
   SELECT P.PATIENTNO, P.PATIENTNAME
   FROM PATIENT P,CONTAINS C
   WHERE P.PATIENTNO = C.PATIENTNO
   AND C.WARDNO = (SELECT WARDNO FROM WARD
   WHERE WARDNAME = 'SURGICAL')

3. List all patients admitted after January 28,2008
   SELECT * FROM PATIENTS P, CONTAINS C
   WHERE P.PATIENTNO = C.PATIENTNO
   AND C.ADMISSIONDATE > '28 JANUARY 2008'

4. Find the names of all the patients being prescribed a drug with name 'Drug112'
   SELECT P.PATNAME FROM PATIENT P, PRESCRIBED PR
   WHERE P.PATIENTNO = PR.PATIENTNO
   AND PR.DRUGNO = (SELECT DRUGNO FROM DRUG
   WHERE DRUGNAME = 'DRUG112')

5. What is the total cost per day of Drug112 supplied to a patient called john smith
   SELECT PR.UNITPERDAY * D.COSTPERUNIT FROM PR.PRESCRIBED, D DRUG
   WHERE PR.DRUGNO = D.DRUGNO
   AND PR.PATIENTNO = (SELECT PATIENTNO FROM PATIENT
   WHERE PATNAME = 'JOHN SMITH')
   AND D.DRUGNO = (SELECT DRUGNO FROM DRUG
   WHERE DRUGNAME = 'DRUG112')

6. Max, min, average number of beds in ward
   SELECT MAX(noOfbeds) as maxNoOfBeds,
   Min(noOfBeds) as minNoOfBeds,
   Avg(noOfBeds) as averageNoOfBeds
   FROM WARD

7. List the patientsNo and names of all patients and the drugno and number of units of their medication. The list should also include the details of patients that are not prescribed any drugs.

```
SELECT P.PATIENTSNO, P.PATNAME, PR.DRUGNO,PR.UNITPERDAY
FROM PATIENT P, PRESCRIBED PR
WHERE P.PATIENTNO = PR.PATIENTNO
GROUP BY P.PATIENTNO
```

CREDITCARD 1

User (userId, firstName, lastName, email, age)
CreditCard (cardNumber, userID, securityCode, expirationDate)
Item (itemId, name, description, pricePerUnit)
Order (orderId, userId, cardNumber, orderTotalAmount)
itemsInOrder (orderId, itemId, quantity)

1. List all users who are greater than 50 years

   ```
   SELECT * FROM USERS
   WHERE AGE > 50
   ```

2. List all credit card information alon with the first and last name of the user

   ```
   SELECT C.* , U.FIRSTNAME, U.LASTNAME
   FROM CREDITCARD C,  USER U
   WHERE C.USERID = U.USERID
   ```

3. List all average pricePerUnit for all items

   ```
   SELECT ITEMID, AVG(PRICEPERUNIT)
   FROM ITEM
   ```

4. List the orderid, ordertotalAmount, the name and description of the items in the order, along with the quantity of the item, sorted by userid and then by itemid.

   ```
   SELECT O.ORDERID, O.ORDERTOTALAMOUNT, I.NAME, I.DESCRIPTION,
   IO.QUANTITY
   FROM ORDER O, ITEM I, ITEMINORDER IO
   WHERE O.ORDERID = IO.ORDERID
   AND I.ITEMID = IO.ITEMID
   ORDER BY O.USERID , I.ITEMID
   ```

5. List how many credicard each user has, along with their email. Only include users who have more than one credit card.

   ```
   SELECT COUNT(C.CARDNUMBER) AS TOTALCREDITCARDS , U.EMAIL
   FROM CREDITCARD C, USER U
   GROUP BY C.EMAIL
   HAVING C.USERID = U.USERID
   AND COUNT(C.CARDNUMBER) >  1
   ```

6. List the userid and cardnumber of all orders, along with the email and age of the users. Include all the users who do not have any orders yet. They have user our telephone operators but have not yet placed any orders.

   ```
   SELECT U.USERID, U.AGE, U.EMAIL, O.CARDNUMBER
   FROM USER U
   LEFT OUTER JOIN ORDER O
   ON O.USERID = U.USERID
   ```

7. List all users firstname and lastname who have the largest age. We may have many users who are the same oldest age. For example we may have three users who are 94 years old.

   ```
   SELECT FIRSTNAME,LASTNAME,AGE
   ORDER BY AGE DESC
   ```

CREDITCARD 2

User (userId, firstName, lastName, email, age)
CreditCard (cardNumber, userID, securityCode, expirationDate)
Item (itemId, name, description, pricePerUnit)
Order (orderId, userId, cardNumber, orderTotalAmount)
itemsInOrder (orderId, itemId, quantity)

1. List email of people which have an expired credit card, the email should be present once even if the user has more expired cards

   ```
   SELECT U.EMAIL FROM USER U, CREDITCARD C
   WHERE U.USERID = C.USERID
   AND C.EXPIRATIONDATE < GETDATE()
   ```

2. List the emails of the user who made the biggest purchase ever. Biggest purchase is defines as largest total amount

   ```
   SELECT EMAIL FROM USER
   WHERE USERID = (SELECT TOP 1 USERID FROM ORDER
                   GROUP BY USERID
                   ORDER BY SUM(ORDERTOTALAMOUNT) DESC)
   ```

Product

Customers(CustomerId, CustomerName,ContactName, Address, City, PostalCode, Country)
Categories(CategoryId, cagtegoryName, Description)
Employees(EmployeeId, LastName, FirstName,BirthDatem Photo, Notes)
OrdersDetails(OrderDetailId, OrderId, ProductId, Quantity)
Orders(OrderId, CustomerId, EmployeeId, OrderDate, ShipperId)
Products(ProductId, ProductName, SupplierId, CategoryId, Unit, Price)
Shippers(ShipperId, ShipperName, Phoen)
Suppliers(SupplierId, SupplierName, ContactName, Address,City,PostalCode,Country, Phone)

1. Create Table OrderDetail
   Create table OrderDetails(OrderDetailId int not null,
   OrderId int, ProductId int, Quantity int,
   PrimaryKey(OrderDetailId)
   ForeignKey(OrderId) references Orders(OrderId),
   ForeignKey(ProductId) reference Products(ProductId))

2. Write a query that will return the productname, unit and price of the product that has sold most units ever. The units can be sold in multiple order. Your query should return one single record or more if equal

```
SELECT PRODUCTNAME, UNIT, PRICE FROM PRODUCT
WHERE PRODUCTID IN (SELECT PRODUCTID FROM ORDERDETAILS
                    GROUP BY PRODUCTID HAVING SUM(QUANTITY)
                      IN(SELECT TOP 1 SUM(QUANTITY) FROM ORDER
                          ORDERDETAIL GROUP BY PRODUCTID ORDER
                            BY SUM(QUANTITY) DESC)
```

3. Write a query that will return the shipper information of the shipper that shipped most number of orders. Return just one record or more if they are equal

```
SELECT * FROM SHIPPER
WHERE SHIPPERID IN (SELECT SHIPPERID FROM ORDERS
                    GROUP BY SHIPPERID
                    HAVING COUNT(*) IN ( SELECT TOP 1 COUNT(*)
                                         FROM ORDER
                                         GROUP BY SHIPPERID
                                         ORDER BY COUNT(*) DESC
```

4. Write a query that will return the employee id, firstname and alstname of the employee which has brought in the most revenue. Revenue is the total money across all orders sold by an employee

```
SELECT EMPLOYEEID,FIRSTNAME,LASTNAME FROM EMPLOYEE
WHERE EMPLOYEEID IN (SELECT TOP 1 O.EMPLOYEEID
                     FROM ORDERS.O, PRODUCTS P, ORDERDETAILS OD
                      WHERE PR.PRODUCTID = OD.PRODUCTID
                      AND O.ORDERID = OD.ORDERID
                      ORDER BY (PR.PRICE*OD.QUANTITY) DESC;
```