PREDICTIVE ANALYTICS AND INVENTORY MANAGEMENT

SQL CODE:

```
            CREATE TABLE Genres
(
 GenreID NUMBER(2) PRIMARY KEY,
 GenreName VARCHAR2(20) NOT NULL UNIQUE
);

INSERT INTO Genres VALUES(3, 'Devotional');
INSERT INTO Genres VALUES(4, 'Romcom');
INSERT INTO Genres VALUES(5, 'SciFi');
INSERT INTO Genres VALUES(6, 'Comedy');
INSERT INTO Genres VALUES(7, 'Horror');
INSERT INTO Genres VALUES(8, 'Thriller');
INSERT INTO Genres VALUES(9, 'Romantic');
INSERT INTO Genres VALUES(10, 'Drama');
INSERT INTO Genres VALUES(11, 'Fantasy');
INSERT INTO Genres VALUES(12, 'Action');
INSERT INTO Genres VALUES(13, 'Crimethriller');
INSERT INTO Genres VALUES(14, 'Technothriller');

CREATE TABLE Stores
(
 StoreID NUMBER(3) PRIMARY KEY,
 Location VARCHAR2(30)
);

INSERT INTO Stores VALUES(1, 'Chicago East');
INSERT INTO Stores VALUES(2, 'Chicago West');
INSERT INTO Stores VALUES(3, 'Chicago Central');
INSERT INTO Stores VALUES(4, 'Chicago Nort');
INSERT INTO Stores VALUES(5, 'Chicago South');

CREATE TABLE Customers
(
 CardNo NUMBER(6) PRIMARY KEY,
 FName VARCHAR2 (30),
 LName VARCHAR2 (30),
 Genre1 VARCHAR2(20),
 Genre2 VARCHAR2(20),
 HomeStore NUMBER(3),
 CONSTRAINT genre1_fk FOREIGN KEY (Genre1)
  REFERENCES Genres (GenreName),
 CONSTRAINT genre2_fk FOREIGN KEY (Genre2)
  REFERENCES Genres (GenreName),
 CONSTRAINT storehomeid_fk FOREIGN KEY (HomeStore)
  REFERENCES Stores(StoreID),
 CONSTRAINT card_check CHECK (LENGTH(CardNo) = 6)
);

INSERT INTO Customers VALUES(237186, 'Dave', 'Davies', 'Animation', 'SciFi', 1);
INSERT INTO Customers VALUES(731678, 'Linda', 'Davies', 'Biography', 'Devotional', 2);
INSERT INTO Customers VALUES(879032, 'Sarvesh', 'Kaushik', 'Romantic', 'Comedy', 3);
INSERT INTO Customers VALUES(267812, 'James', 'Papademas', 'Comedy', 'Romcom', 4);
INSERT INTO Customers VALUES(267813, 'Michael', 'Baah', 'Horror', 'Romantic', 5);
INSERT INTO Customers VALUES(986431, 'Steve', 'Smith', 'Drama', 'Comedy', 3);
INSERT INTO Customers VALUES(926431, 'Steve', 'Smith', 'Horror', 'Comedy', 2);
INSERT INTO Customers VALUES(928432, 'Cary', 'Mohan', 'Comedy', 'Technothriller', 4);
INSERT INTO Customers VALUES(928422, 'William', 'George', 'Romantic', 'Devotional', 4);
INSERT INTO Customers VALUES(928444, 'Sam', 'Cowderly', 'Biography', 'Comedy', 4);
INSERT INTO Customers VALUES(928455, 'Andy', 'Jessy', 'Horror', 'Devotional', 4);
```

```
CREATE TABLE Products(
ProdID VARCHAR2(15) PRIMARY KEY,
ProdCost NUMBER(7,2),
ProdName VARCHAR2(30),
ProdType VARCHAR2(1) CHECK (ProdType IN ('M', 'I')),
ProdInv NUMBER(4)
);

INSERT INTO PRODUCTS VALUES (101, 8.99, 'Star Wars: Force Awakens', 'M', 9);
INSERT INTO PRODUCTS VALUES (102, 20.99, 'Gone with the Wind', 'M', 3);
INSERT INTO PRODUCTS VALUES (103, 20.22, 'Acrimony', 'M', 4);
INSERT INTO PRODUCTS VALUES (104, 20.21, 'HateIQ', 'M', 5);
INSERT INTO PRODUCTS VALUES (105, 20.23, 'Godfather', 'I', 6);
INSERT INTO PRODUCTS VALUES (106, 20.12, 'Acrimony', 'I', 7);
INSERT INTO PRODUCTS VALUES (107, 238.00, 'Spiderman', 'I', 4);
INSERT INTO PRODUCTS VALUES (108, 20.12, 'Godfather1', 'I', 7);
INSERT INTO PRODUCTS VALUES (109, 20.12, 'Godfather2', 'I', 7);
INSERT INTO PRODUCTS VALUES (110, 20.12, 'Godfather3', 'I', 7);
INSERT INTO PRODUCTS VALUES (111, 20.12, 'Godfather4', 'I', 7);
INSERT INTO PRODUCTS VALUES (112, 20.12, 'Godfather5', 'I', 7);
INSERT INTO PRODUCTS VALUES (113, 20.12, 'Godfather6', 'I', 7);

CREATE TABLE Movies
(
  ProdID VARCHAR2(15) PRIMARY KEY,
  MovieYear NUMBER(4),
  MovieGenreID NUMBER(2),
  CONSTRAINT genre_fk FOREIGN KEY (MovieGenreID)
    REFERENCES Genres (GenreID),
  CONSTRAINT ProdID FOREIGN KEY (ProdID)
    REFERENCES PRODUCTS (ProdID)
);

INSERT INTO MOVIES VALUES (101, 2015, 1);
INSERT INTO MOVIES VALUES (102, 1977, 2);
INSERT INTO MOVIES VALUES (103, 1972, 4);
INSERT INTO MOVIES VALUES (104, 1979, 5);
INSERT INTO MOVIES VALUES (105, 1980, 3);

CREATE TABLE Items
(
  ProdID VARCHAR2(15) PRIMARY KEY,
  ItemName  VARCHAR2(25),
  ItemPartNo VARCHAR2(20),
  CONSTRAINT prod_id_fk FOREIGN KEY (ProdID)
    REFERENCES Products (ProdID)
);
INSERT INTO Items VALUES ('104', 'Beats by Dr. Dre', 'XY345');
INSERT INTO ITEMS VALUES ('102', 'RockStar Games', 'RD-45-2017');
INSERT INTO ITEMS VALUES ('106', 'Acrimony', 'RD-46-2018');
INSERT INTO ITEMS VALUES ('103', 'RockStar Games', 'RD-45-2017');
INSERT INTO ITEMS VALUES ('105', 'Ubisoft', 'RD-84-2019');

CREATE TABLE AV_Orders
(
  Order_ID NUMBER(4) PRIMARY KEY,
  CardNo NUMBER(6),
  PMT_Method VARCHAR2(10) CONSTRAINT pmt_check
  CHECK (PMT_Method IN ('CASH', 'CHECK', 'CREDIT')),
  Order_Date DATE,
  CONSTRAINT card#_fk FOREIGN KEY (CardNo)
    REFERENCES Customers (CardNo)
);
INSERT INTO AV_ORDERS VALUES (1001, 731678, 'CREDIT','10-MAR-2023');
INSERT INTO AV_ORDERS VALUES (1002, 267812, 'CASH','10-JAN-2025');
INSERT INTO AV_ORDERS VALUES (1003, 267813, 'CHECK','22-APR-2025');
INSERT INTO AV_ORDERS VALUES (1004, 879032, 'CHECK','28-AUG-2022');
INSERT INTO AV_ORDERS VALUES (1006, 267813, 'CHECK','28-AUG-2022');
INSERT INTO AV_ORDERS VALUES (1005, 267813, 'CHECK','28-AUG-2022');
INSERT INTO AV_ORDERS VALUES (1007, 267813, 'CHECK','28-AUG-2022');
INSERT INTO AV_ORDERS VALUES (1008, 879032, 'CHECK','28-AUG-2022');
INSERT INTO AV_ORDERS VALUES (1009, 267813, 'CHECK','28-AUG-2022');
INSERT INTO AV_ORDERS VALUES (1010, 267813, 'CHECK','28-AUG-2022');
INSERT INTO AV_ORDERS VALUES (1011, 267813, 'CHECK','28-AUG-2022');
INSERT INTO AV_ORDERS VALUES (1012, 267813, 'CHECK','28-AUG-2022');
INSERT INTO AV_ORDERS VALUES (1013, 731678, 'CHECK','28-AUG-2022');
INSERT INTO AV_ORDERS VALUES (1014, 267813, 'CREDIT','28-AUG-2022');
INSERT INTO AV_ORDERS VALUES (1015, 731678, 'CREDIT','28-AUG-2022');

CREATE TABLE Order_Invoice
(
```

```
  Order_ID NUMBER(4),
  LineNo NUMBER(2),
  ProdID VARCHAR2(15),
  QTY NUMBER(3),
  CONSTRAINT order_line_invoice_pk PRIMARY KEY(Order_ID, LineNo),
  CONSTRAINT order_id_invoice_fk FOREIGN KEY (Order_ID)
    REFERENCES AV_ORDERS (Order_ID),
  CONSTRAINT prod_id_invoice_fk FOREIGN KEY (ProdID)
    REFERENCES PRODUCTS (ProdID)
);
INSERT INTO ORDER_INVOICE VALUES (1001,1,'104', 1);
INSERT INTO ORDER_INVOICE VALUES (1002,2,'101', 2);
INSERT INTO ORDER_INVOICE VALUES (1003,3,'106', 1);
INSERT INTO ORDER_INVOICE VALUES (1004,4,'105', 2);
INSERT INTO ORDER_INVOICE VALUES (1005,5,'105', 1);
INSERT INTO ORDER_INVOICE VALUES (1006,6,'102', 4);
INSERT INTO ORDER_INVOICE VALUES (1007,7,'103', 4);
INSERT INTO ORDER_INVOICE VALUES (1008,8,'110', 4);
INSERT INTO ORDER_INVOICE VALUES (1009,9,'110', 4);
INSERT INTO ORDER_INVOICE VALUES (1010,10,'105', 4);
INSERT INTO ORDER_INVOICE VALUES (1011,11,'105', 4);
INSERT INTO ORDER_INVOICE VALUES (1012,12,'105', 4);
INSERT INTO ORDER_INVOICE VALUES (1013,13,'102', 4);
INSERT INTO ORDER_INVOICE VALUES (1014,14,'102', 4);
INSERT INTO ORDER_INVOICE VALUES (1015,15,'102', 4);
```

Required Analytics:

Analyze the schema for information that can be useful to the business or enterprise.

Information about Genres:

What are the top ten genres of choice?

Query:

According to the Genre and Order_Invoice analysis we use analytical function to understand the top genre choice.
Knowing about top genre could be important as we could launch new schemes based on this data to increase the sales.

```
Select GenreName from (SELECT g.genrename as GenreName, count(*)
    FROM order_invoice oi INNER JOIN Movies m
    ON oi.prodid = m.prodid
    JOIN genres g
    ON m.moviegenreid = g.genreid
    group by (g.genrename)
    ORDER BY COUNT (*) DESC) where ROWNUM<=10;
```

Output:

Script Output | Query Result | SQL | All Rows Fetched: 10 in 0.013 seconds

	GENRENAME
1	Devotional
2	Biography
3	Thriller
4	Romcom
5	SciFi
6	Animation
7	Drama
8	Technothriller
9	Crimethriller
10	Action

Moreover, the query described below would print the Top Genres with their count to decide which one is having highest preference.

Script Output | Query Result | SQL | All Rows Fetched: 10 in 0.008 seconds

	GENRENAME	COUNT(*)
1	Devotional	7
2	Biography	4
3	Thriller	4
4	Romcom	3
5	SciFi	1
6	Action	1
7	Crimethriller	1
8	Fantasy	1
9	Animation	1
10	Technothriller	1

Information about Customers:

Who are the top ten customers?

Information related to top customers could be useful to identify special promotional offers exclusive to such customers. In turns it will help to increase the Market Credibility and gain more sales.

Query:

```
Select CardNumber, CountTimes from (SELECT Customers.CARDNO as CardNumber,Count(*) as
CountTimes FROM Customers
LEFT JOIN AV_Orders ON Customers.CARDNO = AV_Orders.CARDNO Group By (Customers.CARDNO)
Order By Count(*) desc)where ROWNUM<=10;
```

Output:

Script Output | Query Result | SQL | All Rows Fetched: 10 in 0.035 seconds

	CARDNUMBER	COUNTTIMES
1	267813	11
2	731678	10
3	879032	2
4	237186	1
5	267812	1
6	926431	1
7	928422	1
8	928432	1
9	928444	1
10	928455	1

Now, If we narrow down the search, from the individual card numbers we can get actual customers name as given below,

Query:

SELECT CUSTOMERS.FNAME,CUSTOMERS.LNAME FROM CUSTOMERS WHERE CARDNO IN(267813,731678,928422,928422,928432,928444,928455,986431,267812,237186,926431);

Script Output | Query Result | SQL | All Rows Fetched: 10 in 0.006 seconds

	FNAME	LNAME
1	Dave	Davies
2	James	Papademas
3	Michael	Baah
4	Linda	Davies
5	Steve	Smith
6	William	George
7	Cary	Mohan
8	Sam	Cowderly
9	Andy	Jessy
10	Steve	Smith

What location has the highest number of customers?

Query:

SELECT Location FROM (select STORES.LOCATION as Location, COUNT(*) FROM STORES LEFT JOIN CUSTOMERS ON STORES.STOREID = CUSTOMERS.HOMESTORE GROUP BY STORES.LOCATION ORDER BY COUNT(*) DESC) WHERE ROWNUM =1;

Script Output | Query Result | SQL | All Rows Fetched: 1 in 0.016 seconds

	LOCATION
1	Chicago Nort

Following is list of all locations ordered from highest to lowest customer preferences.

By knowing the store with the highest sale, we can run special promotional offers, provide more prod variety, and extend store operational hours to increase the overall store profit.

select STORES.LOCATION as Location, COUNT(*) FROM STORES LEFT JOIN CUSTOMERS ON STORES.STOREID = CUSTOMERS.HOMESTORE GROUP BY STORES.LOCATION ORDER BY COUNT(*) DESC

Script Output | Query Result | SQL | All Rows Fetched: 5 in 0.01 seconds

	LOCATION	COUNT(*)
1	Chicago Nort	5
2	Chicago Central	2
3	Chicago West	2
4	Chicago South	1
5	Chicago East	1

What is the preferred method of paying?

Method of paying will have significant impact on the store resources. By establishing robust payment mechanism led to excellent customer service and customer satisfaction increasing chances of re-visits by 95%.

Query:

select Pref_Method from (SELECT AV_ORDERS.PMT_METHOD as Pref_Method, COUNT(*) FROM AV_ORDERS GROUP BY AV_ORDERS.PMT_METHOD ORDER BY COUNT(*) DESC) WHERE ROWNUM=1;

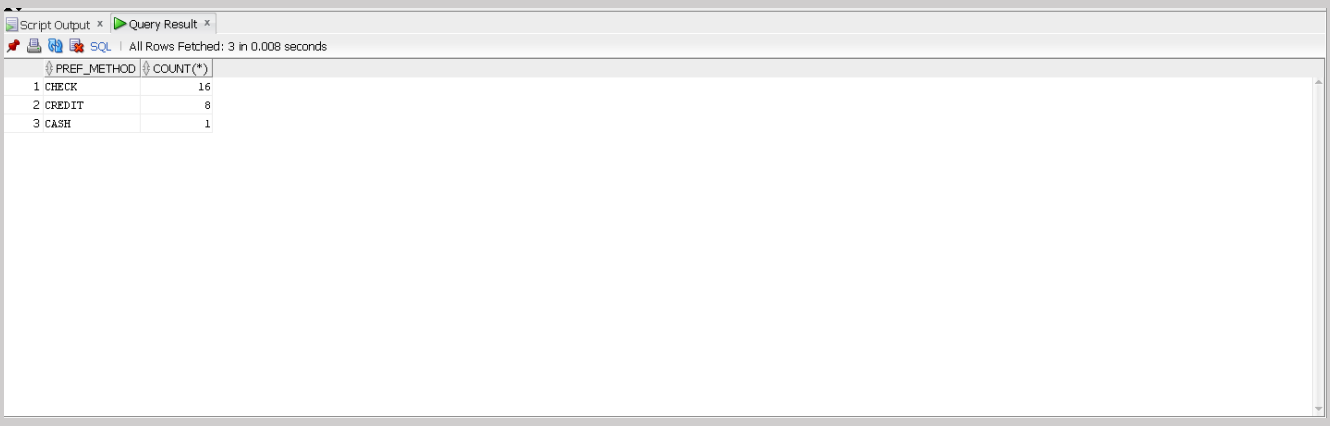
Script Output | Query Result | SQL | All Rows Fetched: 1 in 0 seconds

	PREF_METHOD
1	CHECK

Line 376 Column 163 | Insert | Modified | Windows: CR

Consider the following query demonstrating the popularity of payment method according to the past transactions.

SELECT AV_ORDERS.PMT_METHOD as Pref_Method, COUNT(*) FROM AV_ORDERS GROUP BY AV_ORDERS.PMT_METHOD ORDER BY COUNT(*) DESC;

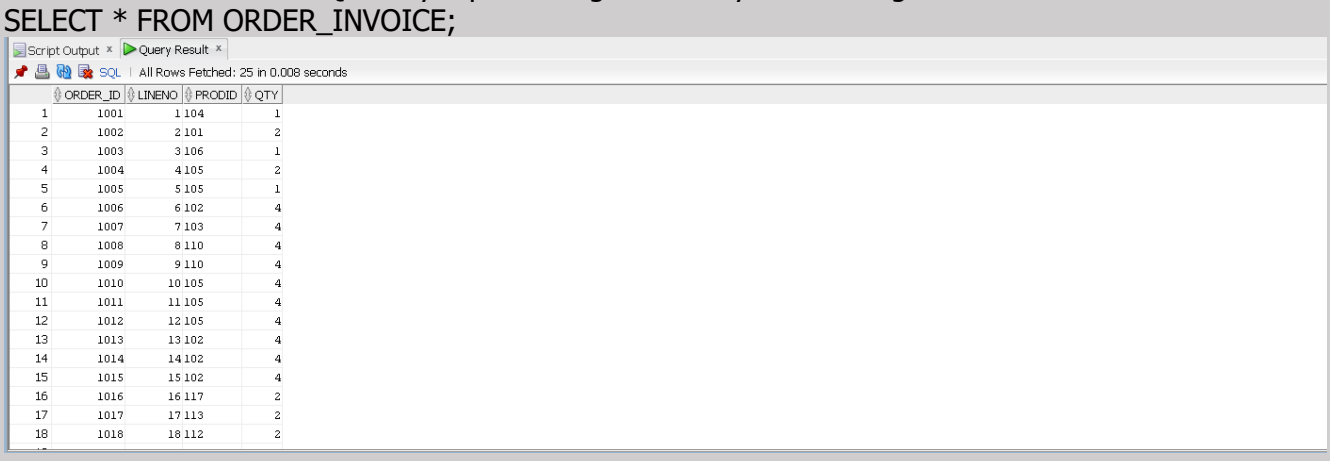

Script Output | Query Result
All Rows Fetched: 3 in 0.008 seconds

	PREF_METHOD	COUNT(*)
1	CHECK	16
2	CREDIT	8
3	CASH	1

What is the average number of rentals?

Consider the following Order_Invoice table where every invoice number represents a rental. Each rental is associated with Qunatity representing how many items being rented.

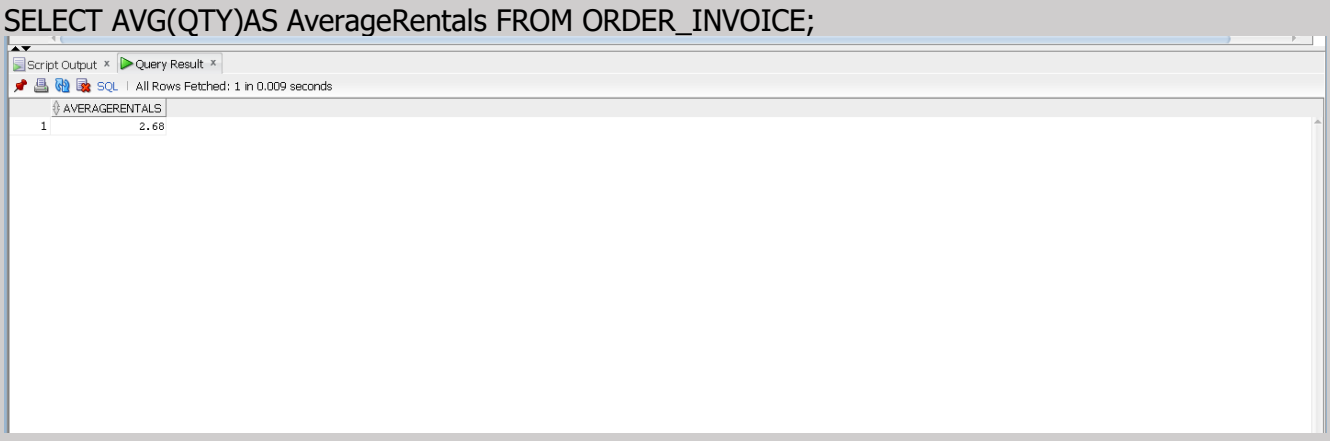
SELECT * FROM ORDER_INVOICE;


Script Output | Query Result
All Rows Fetched: 25 in 0.008 seconds

	ORDER_ID	LINENO	PRODID	QTY
1	1001	1	104	1
2	1002	2	101	2
3	1003	3	106	1
4	1004	4	105	2
5	1005	5	105	1
6	1006	6	102	4
7	1007	7	103	4
8	1008	8	110	4
9	1009	9	110	4
10	1010	10	105	4
11	1011	11	105	4
12	1012	12	105	4
13	1013	13	102	4
14	1014	14	102	4
15	1015	15	102	4
16	1016	16	117	2
17	1017	17	113	2
18	1018	18	112	2

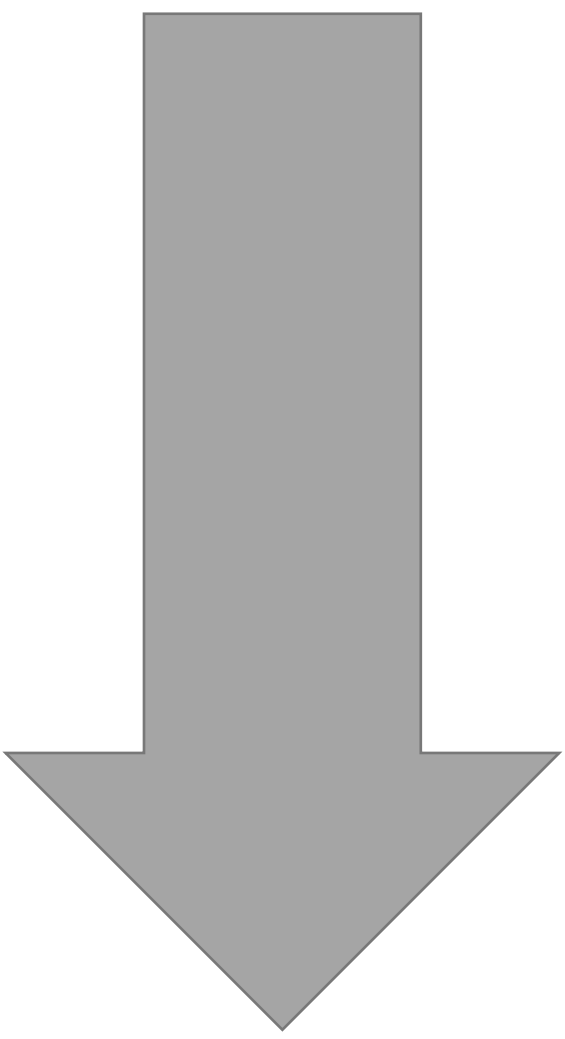
If we take average of all the Quantities in the invoice section it will give us the average rental for all the invoices generated. That in turns will represent the average rentals.

SELECT AVG(QTY)AS AverageRentals FROM ORDER_INVOICE;


Script Output | Query Result
All Rows Fetched: 1 in 0.009 seconds

	AVERAGERENTALS
1	2.68

According to the given analysis We have nearly 2 products rented per rental sale. It an important measure having direct impact on the product inventory cycle. With effective predictive analysis we can step into lean production mechanism. We can predict no of sales and beforehand have the necessary products in stock, eliminating customer loss rate and increasing store revenue.



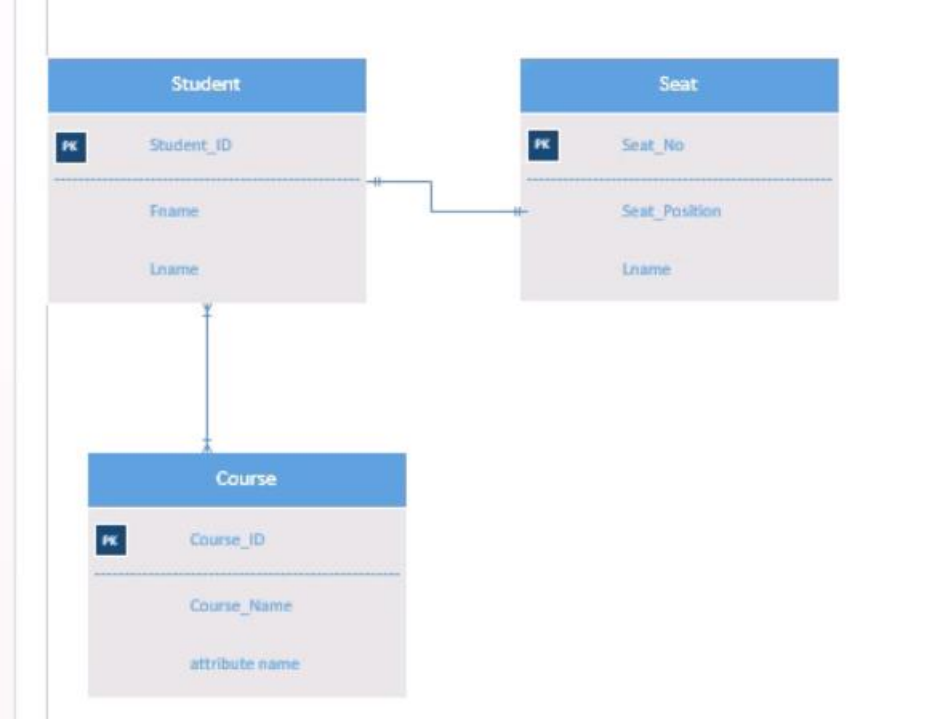
What are the five constraints that Oracle SQL supports for table construction?

Oracle supports the five constraints as given below,

1> Not Null: Not Null is used to identify and accepts only strings with values. as every column field must have a value and cannot be left blank as defined by the NOT NULL.
2> Check: Check Constraint is primarily used for checking a particular condition and only allow the value that satisfies the condition.
3> Unique: Unique constraint is used for ensuring all the values in the column are unique and not repeated.
4> Default: Default keyword is used to specify default value if none of the value is provided for existing field.
5> PRIMARY KEY: Uniquely Identifies each record in a table.
6> FOREIGN KEY: Foreign Key provides link between two tables by creating relationship between the table.

Explain the importance of generating an ERD diagram.

Entity Relationship diagram is having Entities having attributes and representing relationship between them. Entity Relationship Diagram understand what operations organization do and identify the business rules based on the description.
For E.g.:
Consider the following Entity-Relationship Diagram between the Student and the Courses.



How many join conditions are needed to join four tables?

Inner Join Returns the matching value pairs from both the tables it joins the tables and represents the matching entities.
Consider following Tables,

A inner join B inner join C inner join D

If we are about to join the fours tables from A to B we will usually require three inner join statements as each two tables will have relationships within them with the foreign key.

The last query in this lab involves top - n analysis. Define top - n analysis and the ROWNUM identifier.

Top-N analysis is primarily used to fetch the number of rows with some limit from data set which primarily ordered.
ROWNUM is primarily used to limit the number of ROWS retuned by queries. it is called as pseudo column.

How are analytical functions helpful in providing business information?

Analytical functions are helpful while providing the important business-related information and crucial data. If closely look at the first analytical function that we have used to find the top genres in our videorentalstore database we have realize by using the Top-N analysis, ROWNUM, Join and Count Functions we have narrowed down the scope to the desired information.

Similarly, we used various analytical functions in our subsequent queries, helping us to make informed decision about future actions.

Different form of analytics, such a Predictive, Perspective, Descripting, Cognitive and Transitive itself. According to the given business situation we have to identify accurate business problems, once the problems are determined we move forward to identify the root causes and based on that we generate the information with the help of analytical functions.

Such information is really useful while making crucial business decisions related to the Marketing and Analytics.