Stuart School of Business

ILLINOIS INSTITUTE OF TECHNOLOGY

DBMS PROJECT

FOXCORE RETAIL
DESIGNING A DATABASE

MAX 506 – DATABASE DESIGN AND SQL

GROUP 1

ADARSH SABRAD
ADITYARAJ SHARMA
KARAN RANA
PRAVALIKA DASARI

Table of Content

Abstract

1.1 Company Background

Foxcore Retail was founded by Liam Corrigan and Mitchell Fox after they graduated from the University of Western Ontario. The company started as a small retail business selling inexpensive novelty items at music festivals and trade shows. Inspired by a hand-held electric toy and a cooling towel that stayed cool for hours, the two entrepreneurs named their business Foxcore Retail, using a combination of their last names.

As sales grew, the partners hired salespeople to help them handle high demand, paying commissions to incentivize performance. They also tested a variety of new products tailored to specific kinds of shows. By its second year, Foxcore had expanded and was managing up to three shows per weekend, with multiple booths at some venues.

1.2 Business Problem

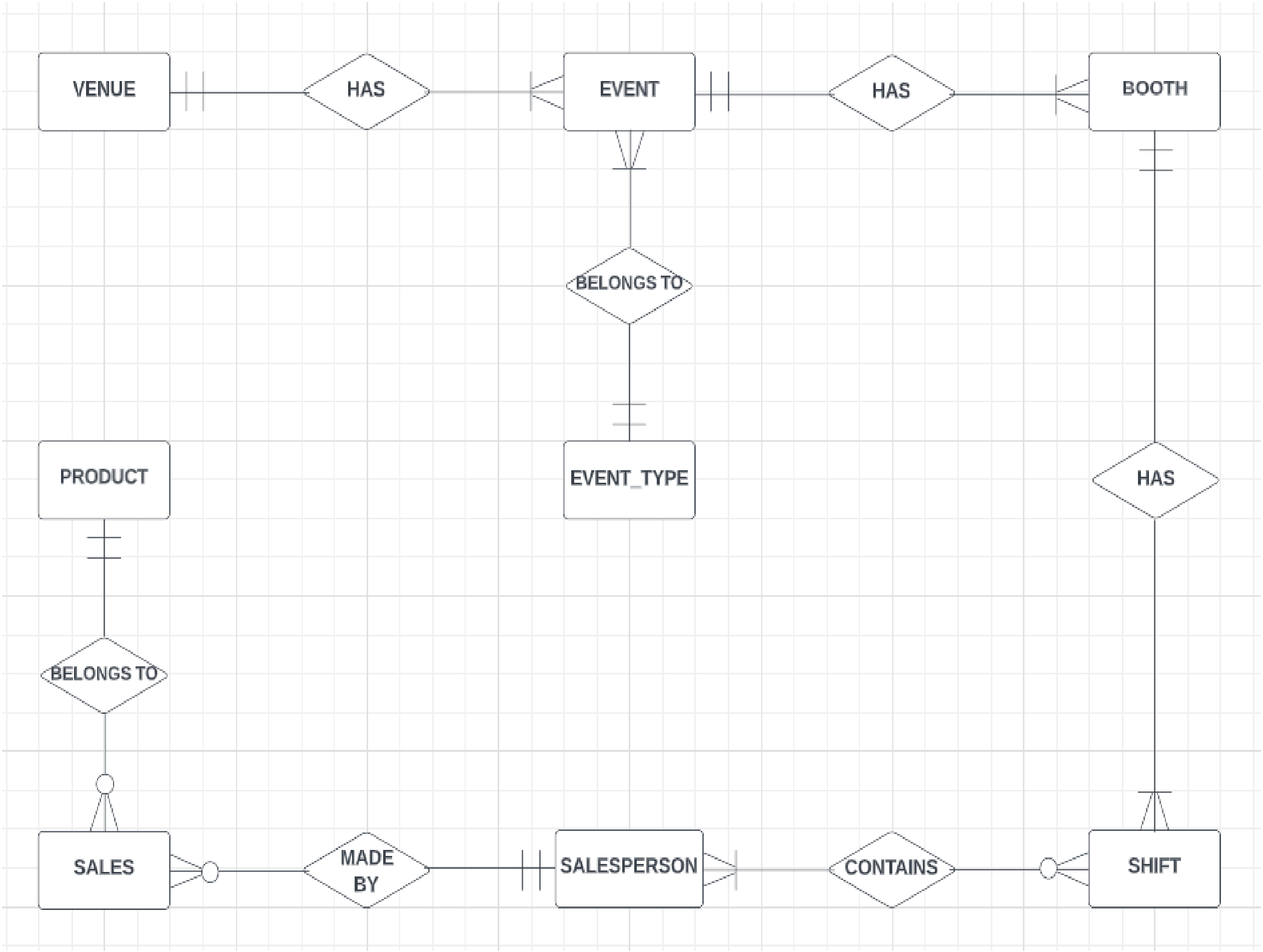
The business problem faced by Corrigan and Fox of Foxcore Retail is the inefficiency and inaccuracy of their manual sales-tracking system. The partners had been using a paper-based system to record sales, which led to errors, lost commissions, and unreliable inventory estimates. As the business grew, the partners realized that they needed to implement a more disciplined database system to collect and analyze valuable data that could provide insight into the performance of employees, events, and products.

The inability to access and analyze the information resulted in sub-optimal decision-making for both short- and long-term planning. The partners recognized that they needed to track exact details about which products were sold, by whom, when, and where. The business problem was to find a solution that would allow them to efficiently and accurately track sales data to improve their operations and decision-making.

1.3 Goal

To address the business problem of inefficiency and inaccuracy in Foxcore Retail's manual sales-tracking system, a custom relational database system will be designed. This system will enable the efficient tracking of events, sales consultants, and individual sales of various products. By accessing and analyzing this database, valuable data can be collected to provide strategic insights that will improve decision-making for both short- and long-term planning. With this database, Foxcore Retail can accurately track their sales, identify which products are selling well, and optimize their operations accordingly.

2. ER Diagram



The relationships that we defined from the ER Diagram are as follows:

VENUE --- EVENT

A venue can host many events, but each event can only take place at one venue.

EVENT --- EVENT_TYPE

An event can have only one event type, but each event type can be assigned to multiple events.

EVENT --- BOOTH

Each event can have multiple booths, but each booth can belong to only one event.

BOOTH --- SHIFT

Each booth can have multiple shifts, but each shift belongs to a booth.

SHIFT --- SALESPERSON

A shift can contain multiple salespersons, but each salesperson can have zero to multiple shifts.

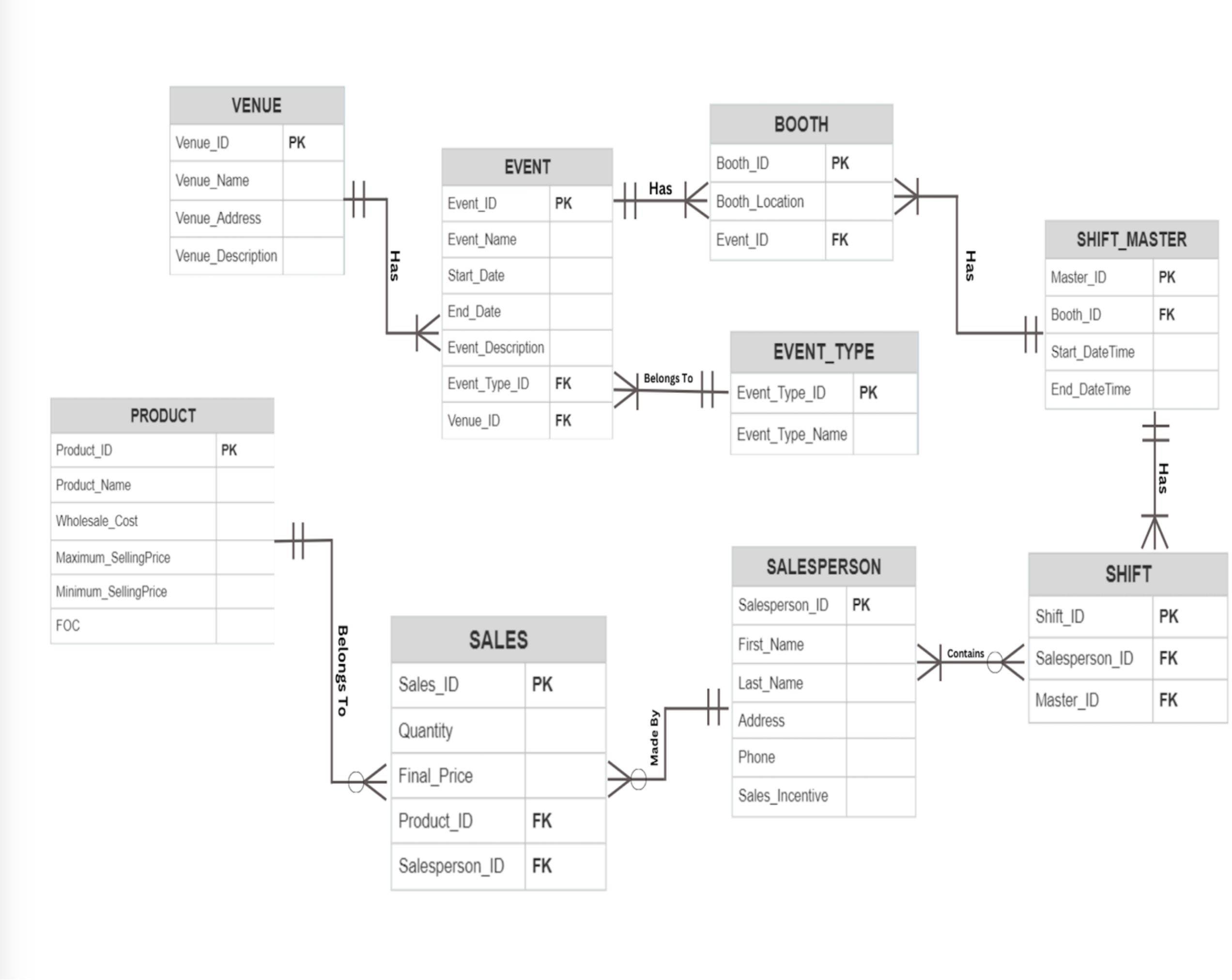
SALESPERSON --- SALES

Each sale is made by one salesperson, but each salesperson can make multiple sales.

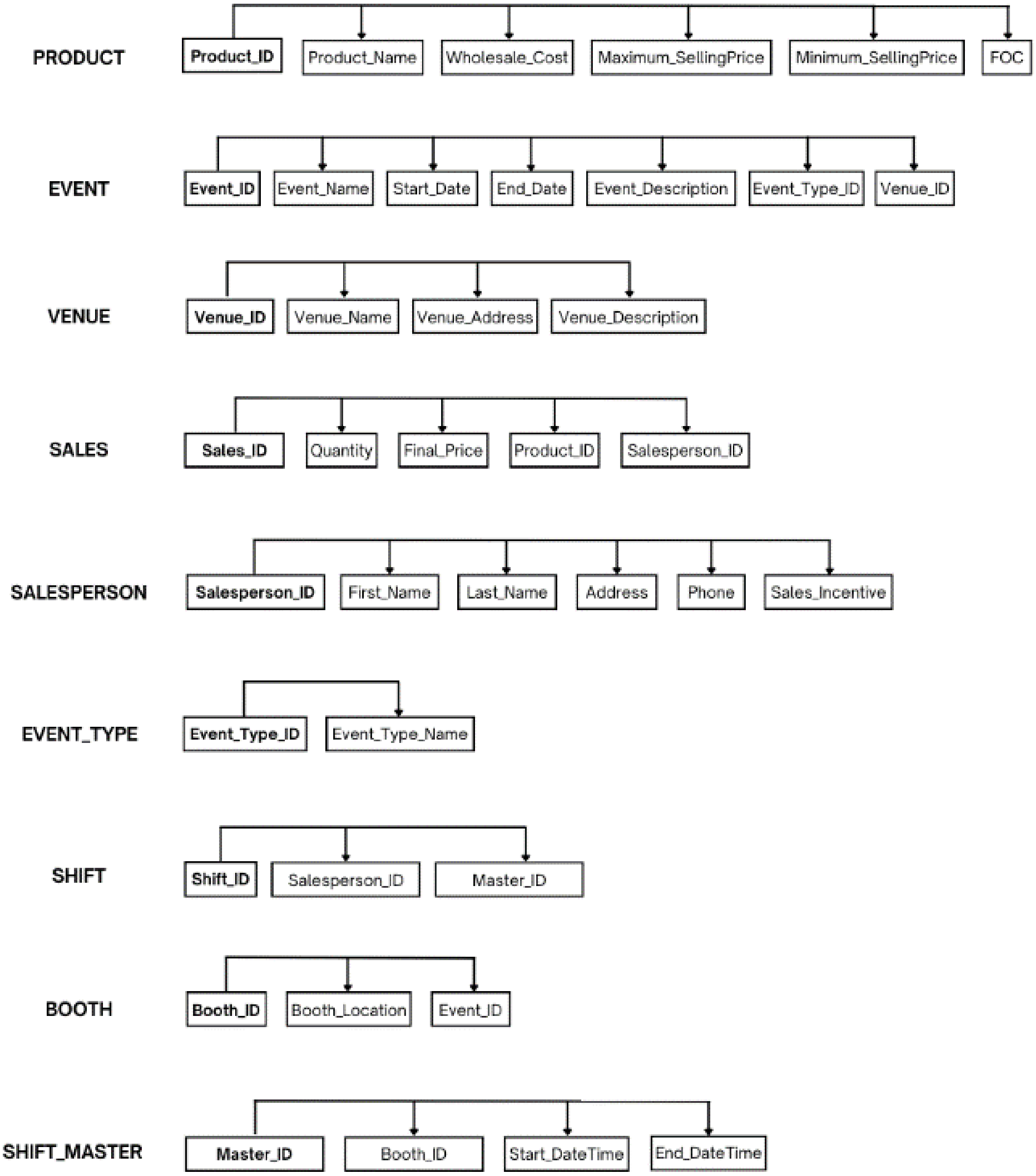
SALES --- PRODUCT

Each sale is associated with only one product, but each product can be associated with multiple sales.

3. Foxcore Retail DB Relational Schema



4. Normalization



PRODUCT: **Product_ID**, Product_Name, Wholesale_Cost, Maximum_SellingPrice, Minimum_SellingPrice, FOC

EVENT: **Event_ID**, Event_Name, Start_Date, End_Date, Event_Description, Event_Type_ID, Venue_ID

VENUE: **Venue_ID**, Venue_Name, Venue_Address, Venue_Description

SALES: **Sales_ID**, Quantity, Final_Price, Product_ID, Salesperson_ID

SALESPERSON: **Salesperson_ID**, First_Name, Last_Name, Address, Phone, Sales_Incentive

EVENT_TYPE: **Event_Type_ID**, Event_Type_Name

SHIFT: **Shift_ID**, Salesperson_ID, Master_ID

BOOTH: **Booth_ID**, Booth_Location, Event_ID

SHIFT_MASTER: **Master_ID**, Booth_ID, Start_DateTime, End_DateTime

5. DATABASE CREATION: DDL-SQL Command

Entity	Attributes	Datatype
VENUE	**Venue_ID (PK)**	VARCHAR(10)
	Venue_Name	VARCHAR(50)
	Venue_Address	VARCHAR(100)
	Venue_Description	VARCHAR(200)
EVENT	**EVENT_ID (PK)**	VARCHAR(10)
	Event_Name	VARCHAR(50)
	Start_Date	DATE
	End_Date	DATE
	Event_Description	VARCHAR(200)
BOOTH	**Booth_ID (PK)**	VARCHAR(10)
	Booth_Location	VARCHAR(200)
EVENT_TYPE	**Event_Type_ID (PK)**	VARCHAR(10)
	Event_Type_Name	VARCHAR(50)
SHIFT_MASTER	**Master_ID (PK)**	VARCHAR(10)
	Start_DateTime	DATETIME
	End_DateTime	DATETIME
SHIFT	**Shift_ID (PK)**	VARCHAR(10)
SALESPERSON	Salesperson_ID	VARCHAR(10)
	First_Name	VARCHAR(50)
	Last_Name	VARCHAR(50)
	Address	VARCHAR(200)
	Phone	VARCHAR(20)
	Sales_Incentive	DECIMAL(5,2)
SALES	**Sales_ID (PK)**	VARCHAR(10)
	Quantity	INTEGER
	Final_Price	DECIMAL(10,2)
PRODUCT	**Product_ID (PK)**	VARCHAR(10)
	Product_Name	VARCHAR(50)
	Wholesale_Cost	DECIMAL(10,2)
	Maximum_SellingPrice	DECIMAL(10,2)
	Minimum_SellingPrice	DECIMAL(10,2)
	FOC	BIT

CREATE SCHEMA [FoxcoreDatabase | AUTHORIZATION FoxcoreOwner]

VENUE

```
CREATE TABLE VENUE (
   Venue_ID VARCHAR(10) NOT NULL,
   Venue_Name VARCHAR(50) NOT NULL,
   Venue_Address VARCHAR(100) NOT NULL,
   Venue_Description VARCHAR(200),
   PRIMARY KEY (Venue_ID)
);
```

EVENT

```
CREATE TABLE EVENT (
   Event_ID VARCHAR(10) NOT NULL ,
   Event_Name VARCHAR(50) NOT NULL,
   Start_Date DATE NOT NULL,
   End_Date DATE NOT NULL,
   Event_Description VARCHAR(200),
   Event_Type_ID VARCHAR(10) NOT NULL,
   Venue_ID VARCHAR(10) NOT NULL,
   PRIMARY KEY (Event_ID),
   FOREIGN KEY (Event_Type_ID) REFERENCES EVENT_TYPE(Event_Type_ID),
   FOREIGN KEY (Venue_ID) REFERENCES VENUE(Venue_ID)
);
```

BOOTH

```
CREATE TABLE BOOTH (
   Booth_ID VARCHAR(10) NOT NULL,
   Booth_Location VARCHAR(20) NOT NULL,
   Event_ID VARCHAR(10) NOT NULL,
   PRIMARY KEY (Booth_ID),
   FOREIGN KEY (Event_ID) REFERENCES EVENT(Event_ID)
);
```

EVENT TYPE

```
CREATE TABLE EVENT_TYPE (
   Event_Type_ID VARCHAR(10) NOT NULL,
   Event_Type_Name VARCHAR(50) NOT NULL,
   PRIMARY KEY (Event_Type_ID)
);
```

SHIFT MASTER

```
CREATE TABLE SHIFT_MASTER (
   Master_ID VARCHAR(10) NOT NULL,
   Booth_ID VARCHAR(10) NOT NULL,
   Start_DateTime DATETIME NOT NULL,
   End_DateTime DATETIME NOT NULL,
   PRIMARY KEY (Master_ID),
   FOREIGN KEY (Booth_ID) REFERENCES BOOTH(Booth_ID)
);
```

SHIFT

```
CREATE TABLE SHIFT (
   Shift_ID VARCHAR(10) NOT NULL,
   Master_ID VARCHAR(10) NOT NULL,
   Salesperson_ID VARCHAR(10) NOT NULL,
   PRIMARY KEY (Shift_ID),
   FOREIGN KEY (Salesperson_ID) REFERENCES SALESPERSON(Salesperson_ID),
   FOREIGN KEY (Master_ID) REFERENCES SHIFT_MASTER(Master_ID)
);
```

SALESPERSON

```
CREATE TABLE SALESPERSON (
   Salesperson_ID VARCHAR(10) NOT NULL,
   First_Name VARCHAR(50) NOT NULL,
   Last_Name VARCHAR(50) NOT NULL,
   Address VARCHAR(200) NOT NULL,
   Phone VARCHAR(20) NOT NULL
   CHECK (Phone LIKE ‘___-____’),
   Sales_Incentive DECIMAL(5,2) ,
   CHECK (Sales_Incentive BETWEEN 0 AND 100),
   PRIMARY KEY (Salesperson_ID)
);
```

SALES

```
CREATE TABLE SALES (
   Sales_ID VARCHAR(10) NOT NULL,
   Quantity INTEGER NOT NULL,
   Final_Price DECIMAL(10, 2) NOT NULL,
   Product_ID INTEGER NOT NULL,
   Salesperson_ID INTEGER NOT NULL,
   PRIMARY KEY (Sales_ID),
   FOREIGN KEY (Product_ID) REFERENCES PRODUCT(Product_ID),
   FOREIGN KEY (Salesperson_ID) REFERENCES SALESPERSON(Salesperson_ID)
);
```

PRODUCT

```
CREATE TABLE PRODUCT (
   Product_ID VARCHAR(10) NOT NULL,
   Product_Name VARCHAR(50) NOT NULL,
   Wholesale_Cost DECIMAL(10, 2) NOT NULL,
   Maximum_SellingPrice DECIMAL(10, 2) NOT NULL,
   Minimum_SellingPrice DECIMAL(10, 2) NOT NULL,
   FOC BIT NOT NULL Default 0,
   PRIMARY KEY (Product_ID)
);
```

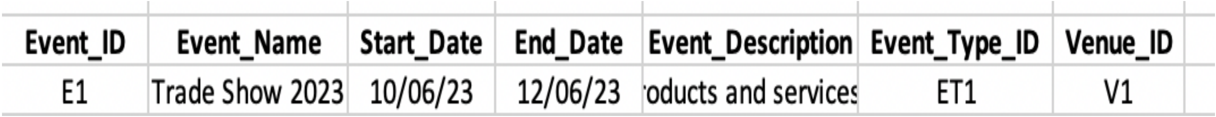
6. DML

We would use the SQL INSERT, UPDATE, and DELETE commands to populate and modify the current database. The following are the sample statements that could be used to help Foxcore Retail ensure its database is up-to-date and kept clear of unnecessary or outdated information.

The following are the sample statements that could be used to help Foxcore Retail ensure its database is up-to-date and kept clear of unnecessary or outdated information.

1. **To Insert a new row about all the details information on Event.**

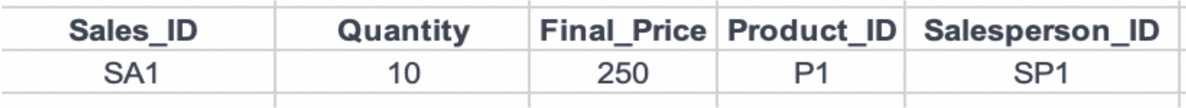
```
INSERT INTO EVENT (Event_ID, Event_Name, Start_Date, End_Date, Event_Description,
Event_Type_ID, Venue_ID) VALUES (‘E1’, ‘Trade Show 2023’, ‘2023-06-10’,
‘2023-06-12’, ‘Come and see the latest products and services from the industry leaders.’,
‘ET1’, ‘V1’);
```



Event_ID	Event_Name	Start_Date	End_Date	Event_Description	Event_Type_ID	Venue_ID
E1	Trade Show 2023	10/06/23	12/06/23	oducts and services	ET1	V1

2. **To Insert a new row containing information on Sales from the Event.**

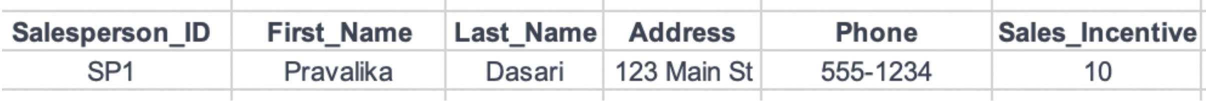
```
INSERT INTO SALES (Sales_ID, Quantity, Final_Price, Product_ID, Salesperson_ID)
VALUES (‘SA1’, 10, 250.00, ‘P1’, ‘SP1’);
```



Sales_ID	Quantity	Final_Price	Product_ID	Salesperson_ID
SA1	10	250	P1	SP1

3. To Insert information about the Salesperson working at the Event.

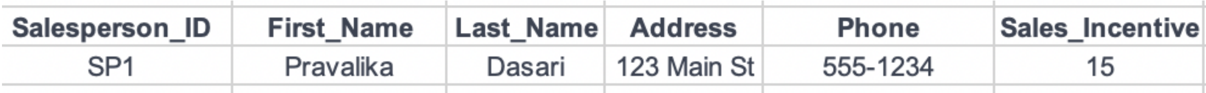
```
INSERT INTO SALESPERSON (Salesperson_ID, First_Name, Last_Name, Address, Phone,
Sales_Incentive) VALUES (‘SP1’,‘Pravalika’, ‘Dasari’, ‘123 Main St’,‘555-1234’, 10);
```



Salesperson_ID	First_Name	Last_Name	Address	Phone	Sales_Incentive
SP1	Pravalika	Dasari	123 Main St	555-1234	10

4. How can you update the Sales_Incentive in Salesperson table from 10.00 to 15.00?

```
UPDATE SALESPERSON
SET Sales_Incentive = 15.00
WHERE Salesperson_ID = ‘SP1’;
```

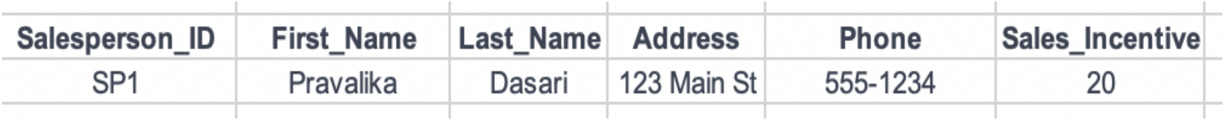


Salesperson_ID	First_Name	Last_Name	Address	Phone	Sales_Incentive
SP1	Pravalika	Dasari	123 Main St	555-1234	15

5. How can you update the Sales_Incentive in Salesperson table who has sales greater than 200?

```
UPDATE Salesperson
SET Sales_Incentive = 20.00
WHERE Salesperson.ID IN (
   SELECT Salesperson_ID
   FROM Sales
```

```
    WHERE Final_Price > 200
);
```



Salesperson_ID	First_Name	Last_Name	Address	Phone	Sales_Incentive
SP1	Pravalika	Dasari	123 Main St	555-1234	20

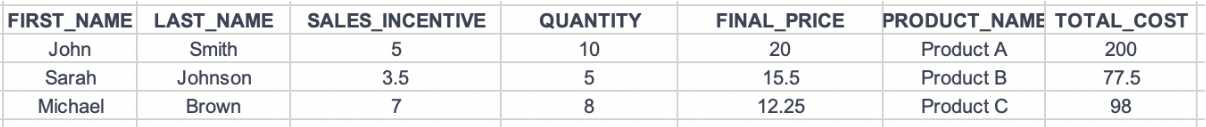
6. How to Delete Product whose Product_ID is P1?

```
DELETE FROM Product WHERE Product_ID = ‘P1’;
```

REPORTS

1. Create a report that displays information about each salesperson, the percentage of incentive they earned, and the quantity of each product they sold. Additionally, include the name and selling price of each product, and the total sales for each product.

```
SELECT S.FIRST_NAME, S.LAST_NAME, S.SALES_INCENTIVE, ST.QUANTITY,
ST.FINAL_PRICE, P.PRODUCT_NAME, (ST.QUANTITY * ST.FINAL_PRICE) AS TOTAL_COST
FROM SALESPERSON S
INNER JOIN SALES ST ON S.SALESPERSON_ID = ST.SALESPERSON_ID
INNER JOIN PRODUCT P ON S.PRODUCT_ID = P. PRODUCT _ID
```

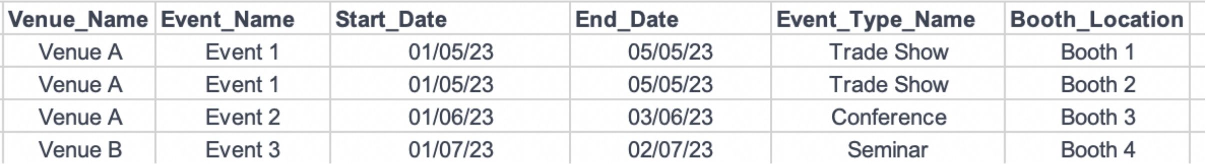


FIRST_NAME	LAST_NAME	SALES_INCENTIVE	QUANTITY	FINAL_PRICE	PRODUCT_NAME	TOTAL_COST
John	Smith	5	10	20	Product A	200
Sarah	Johnson	3.5	5	15.5	Product B	77.5
Michael	Brown	7	8	12.25	Product C	98

2. Display a list of all events along with their corresponding details.

```
Select Venue_Name, Event_Name, Start_Date,End_Date,Event_Type_Name,
Booth_Location
From event as E
```

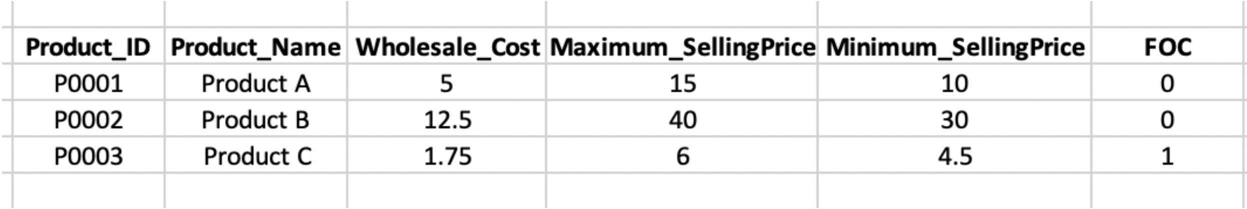
Inner join Event_Type ET on E.Event_Type_id = ET.EVENT_TYPE_ID
INNER JOIN VENUE V ON E.VENUE_ID = V.VENUE_ID
INNER JOIN BOOTH B ON E.EVENT_ID = B.EVENT_ID



Venue_Name	Event_Name	Start_Date	End_Date	Event_Type_Name	Booth_Location
Venue A	Event 1	01/05/23	05/05/23	Trade Show	Booth 1
Venue A	Event 1	01/05/23	05/05/23	Trade Show	Booth 2
Venue A	Event 2	01/06/23	03/06/23	Conference	Booth 3
Venue B	Event 3	01/07/23	02/07/23	Seminar	Booth 4

3. List of Products sold by Foxcore.

Select Product_ID, Product_Name, Wholesale_Cost, Maximum_SellingPrice, Minimum_SellingPrice
From Product



Product_ID	Product_Name	Wholesale_Cost	Maximum_SellingPrice	Minimum_SellingPrice	FOC
P0001	Product A	5	15	10	0
P0002	Product B	12.5	40	30	0
P0003	Product C	1.75	6	4.5	1

Conclusion

There was inefficiency and inaccuracy in Foxcore Retails manual sales-tracking system. The solution to this problem was to design a custom relational database system that would enable the efficient tracking of events, sales consultants, and individual sales of various products. By accessing and analysing this database, valuable data could be collected to provide strategic insights that would improve decision-making for both short- and long-term planning. With this database, Foxcore Retail can accurately track its sales, identify which products are selling well, and optimize its operations accordingly.

In conclusion, the report outlines a comprehensive solution to the business problem faced by Foxcore Retail and provides a detailed plan for implementing a custom relational database system. The proposed database system is expected to improve the efficiency and accuracy of sales tracking, thereby enabling Foxcore Retail to make informed decisions about their operations and optimize their performance.