FoxCore Retail Database Design Project

GROUP 4 MEMBERS

Vinutha Chelamallu

Hsiao Tien (Tammy) Chow

Somya Garg

Ta-Chin (Meek) Ho

Sai Pradeep Vallabhaneni

Agenda

- **Background**
  - Origin and Growth
  - Challenges
  - Constraints
- **Creation of Database**
  - Conceptual
  - Logical
  - Physical
- **Scripts for Database Design**

Origin and Growth of FoxCore

- A small retail business across Ontario founded by Liam Corrigan and Mitchell Fox.
- Selling inexpensive novelty items : Bubble Guns and Arctic Skin Cooling Towels, which Fox had previously built a relationship with the owner.
- As sales grew, they hired salespeople to handle the high demand and added new products such as emoji pillows, heating pads, and remote-controlled drones, while figuring out which products worked best at different shows.
- Foxcore had expanded. Corrigan and Fox managed up to three shows per weekend, with multiple booths at some venues.

Challenges

Data Fragmentation

Information spread across multiple spreadsheets made routine business operations difficult and lacked strategic insights.

Data Analysis Limitations

Inability to access and analyze essential data hindered optimal decision-making for both immediate and future planning.

Inaccurate Sales Tracking

Consultants failed to record all sales, leading to lost commissions and unreliable inventory estimates due to unclear sales records.

Constraints

Time

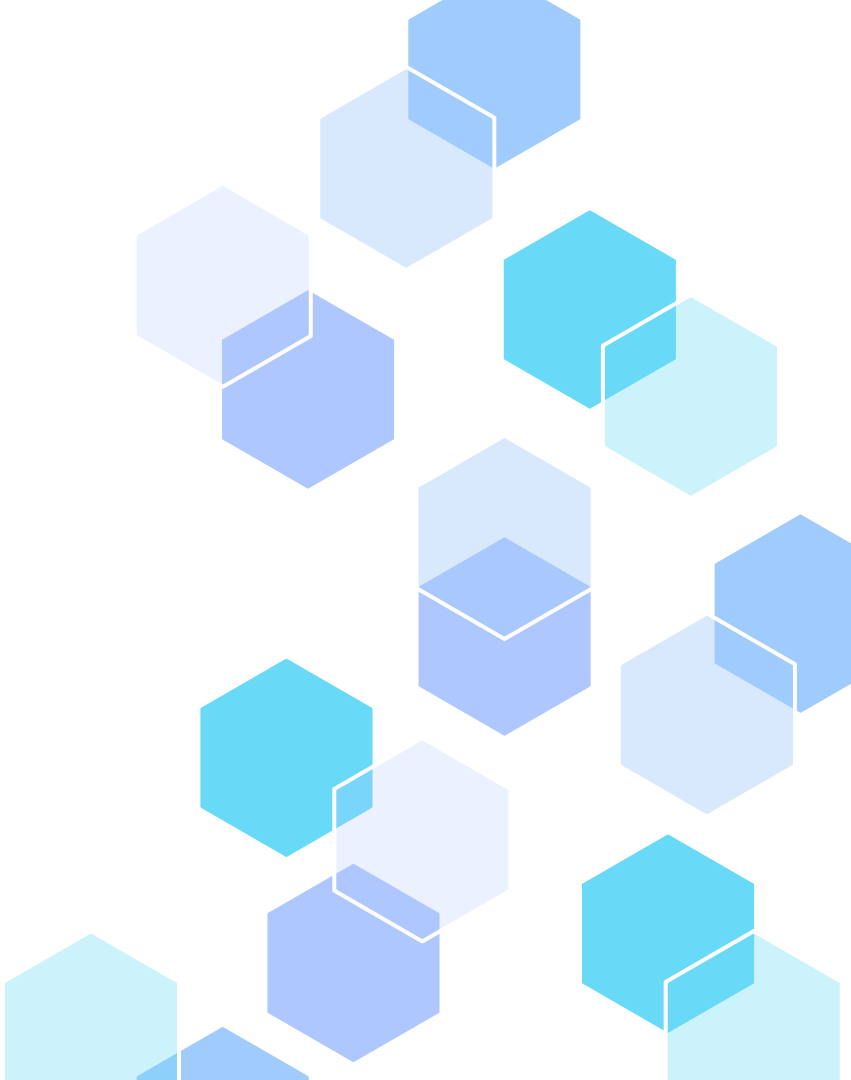
Required a quick and effective solution in less than 30 days before the festival season began

Budget

Foxcore struggled to set appropriate costs and prices due to rising expenses and needed efficient, cost-effective solutions quickly.

Let’s Create a DataBase!

- **Focus:** Designed to track events, sales, salespersons, venues, products, and booths.
- **Purpose:** Protect business records from constraints and ensure operational efficiency.



Conceptual Database Design

Identify Entities and Relationships: Define the entities and relationships in the database.

Attribute Listing: List attributes for each entity to structure the data effectively.

Reduce Redundancies and Inconsistencies: A well-designed conceptual database minimizes data redundancies and inconsistencies.

Description & Relationships of attributes

Entity	Attributes	Description	Relationships
Venue	**Venue ID(primary key)**,Venue Name,Street Address,City,State/Province,Postal Code,Description	Information about the venue where events are held.	One Venue can host Many Events
Event	**Event ID(primary key), Venue ID(foreign key)**,Event Name,StartDate,EndDate,Description,EventType	Details about the event.	One Event can have Many Booths One Event can have Many Sales
Booth	**Booth ID(primary key), Event ID(foreign key)**,Location	Information about a booth within a specific event.	One Booth can have Many Sales
Product	**Product ID(primary key)**,ProductName,WholeSale Code,Minimum Selling Price	Details about a product sold at events.	One Product can be sold in Many Sales
SalesPerson	**SalesPerson ID(primary key),** FirstName,LastName,Street Address,City,State/Province,Postal Code,Phone Number	Information about the salesperson	One Salesperson can work Many Shifts (Sales) One Salesperson can work many Shifts
Sales	**Sales ID(primary key)**,**Event ID(foreign key)**,**Shift ID(foreign key)**,**Product ID(foreign key)**,Quantity Sold,Selling Price,Sale Date	Details about a specific sales transaction	Associates product and salesperson
Shift	**Shift ID(primary key)**,**SalesPerson ID(foreign key),** **Booth ID(foreign key)**,StartTime,EndTime	Information about a salesperson's work schedule at a specific booth during an event.	One Shift is assigned to one Booth

Primary Key & Foreign Key
(Enity : Venue / Product / Event / Salesperson)

Entity	Attributes	Primary Key	Foreign Key	Data Type
Venue	VenueID	VenueID		VARCHAR(10)
	VenueName			CHAR(20)
	Street Address			VARCHAR(100)
	City			VARCHAR(25)
	State/Province			VARCHAR(15)
	PostalCode			INT(5)
	Description			VARCHAR(255)

Entity	Attributes	Primary Key	Foreign Key	Data Type
Product	ProductID	ProductID		VARCHAR(10)
	ProductName			VARCHAR(45)
	WholeSaleCode			DECIMAL(10,2)
	MinimumSellingPrice			DECIMAL(10,2)

Entity	Attributes	Primary Key	Foreign Key	Data Type
Event	EventID	EventID		VARCHAR(10)
	VenueID		VenueID **(References Venue.VenueID)**	VARCHAR(45)
	EventName			DATE
	StartDate			DATE
	EndDate			VARCHAR(255)
	Description			VARCHAR(10)
	EventType			VARCHAR(25)

Entity	Attributes	Primary Key	Foreign Key	Data Type
SalesPerson	SalesPersonID	SalesPersonID		VARCHAR(10)
	FirstName			VARCHAR(25)
	LastName			VARCHAR(25)
	Street Address			VARCHAR(100)
	City			VARCHAR(25)
	State/Province			VARCHAR(15)
	PostalCode			INT(6)
	PhoneNumber			INT(10)

Primary Key & Foreign Key (Enity : Sales/ Booth/Shift)

Entity	Attributes	Primary Key	Foreign Key	Data Type
Sales	SalesID	SalesID		VARCHAR(10)
	EventID		EventID **(References Event.EventID)**	VARCHAR(10)
	ShiftID		ShiftID **(References Shift.ShiftID)**	VARCHAR(10)
	ProductID		**ProductID (References Product.ProductID)**	VARCHAR(10)
	QuantitySold			INT
	SellingPrice			DECIMAL(10,2)
	SaleDate			DATE

Entity	Attributes	Primary Key	Foreign Key	Data Type
Booth	BoothID	BoothID		VARCHAR(10)
	EventID		EventID **(References Event.EventID)**	VARCHAR(10)
	Location			VARCHAR(255)

Entity	Attributes	Primary Key	Foreign Key	Data Type
Shift	ShiftID	ShiftID		VARCHAR(10)
	SalesPerson ID		SalesPersonID **(References SalesPerson.SalesPersonID)**	VARCHAR(10)
	BoothID		BoothID **(References Booth.BoothID)**	VARCHAR(10)
	StartTime			TIME
	EndTime			TIME

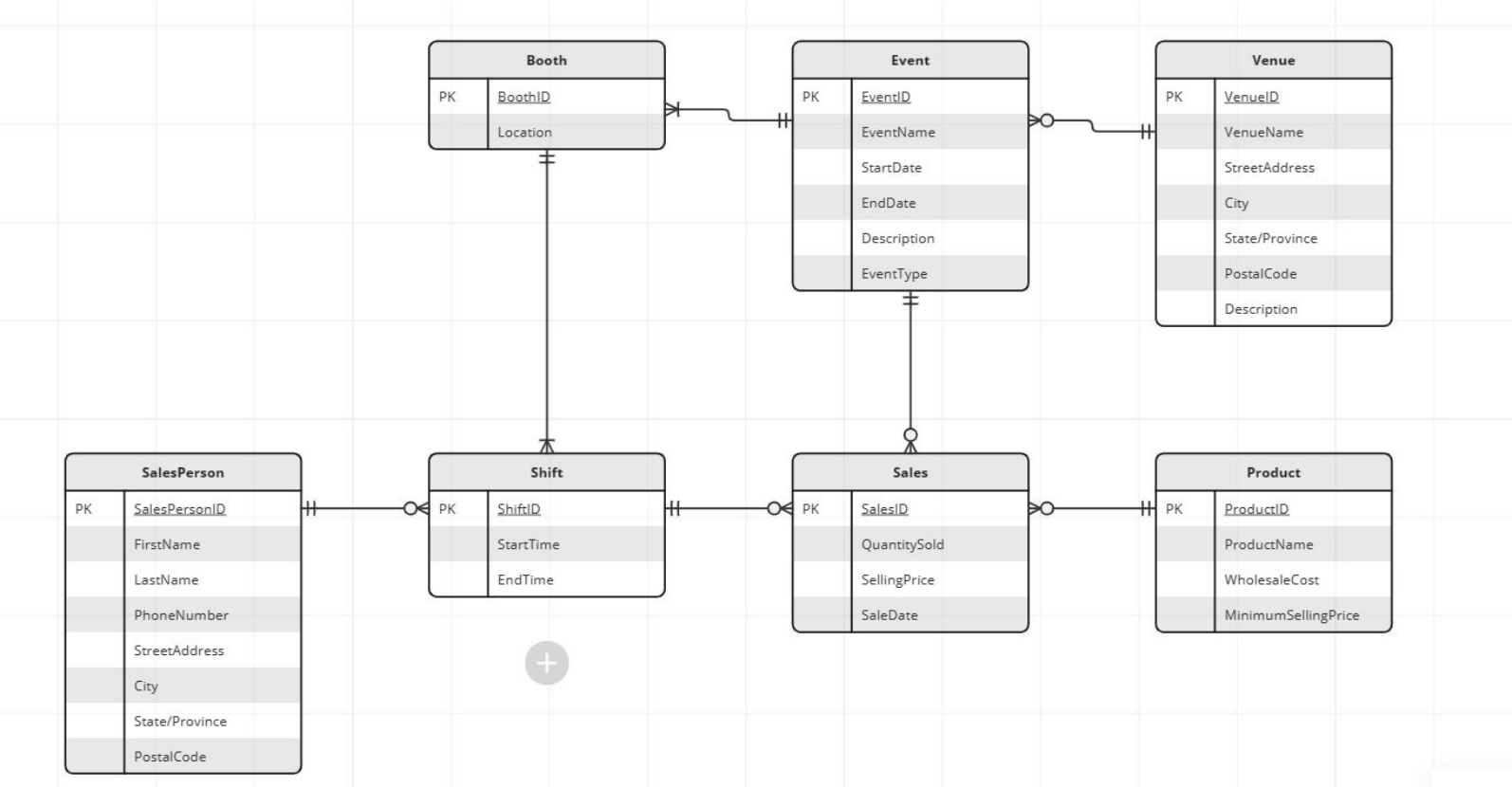
Logical Database Design

This can be the part of the presentation where you introduce yourself, write your email...

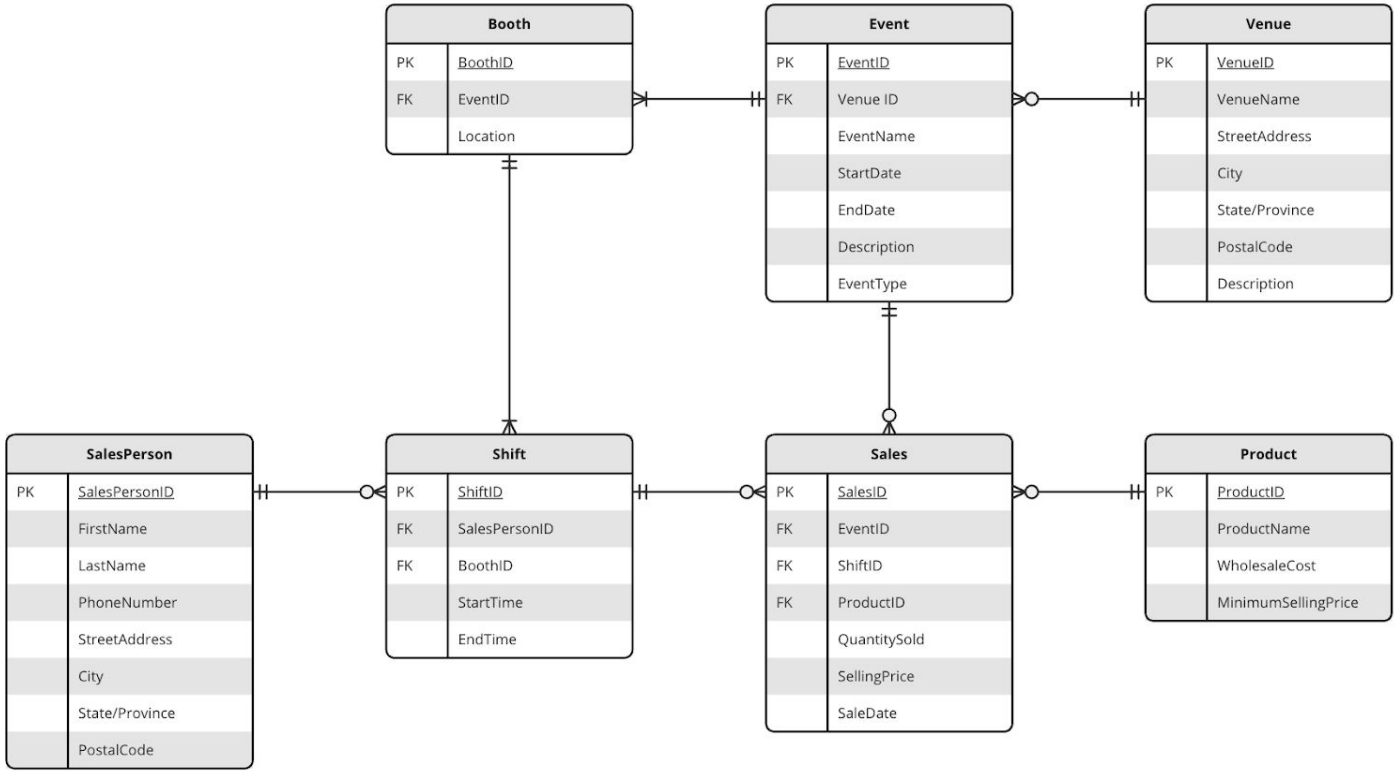
Normalization

Identify	Ensure	Check	Review
Identify entities, relationships, and attributes	Ensure the relations support required transactions	Check integrity constraints	Review logical data model with user

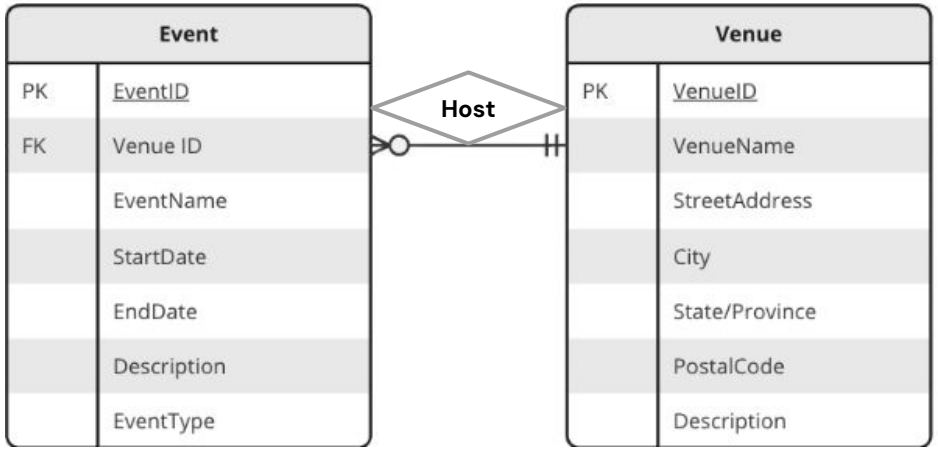
ER DIAGRAM


Booth
PK
BoothID
Location
Event
PK
EventID
EventName
StartDate
EndDate
Description
EventType
Venue
PK
VenueID
VenueName
StreetAddress
City
State/Province
PostalCode
Description
SalesPerson
PK
SalesPersonID
FirstName
LastName
PhoneNumber
StreetAddress
City
State/Province
PostalCode
Shift
PK
ShiftID
StartTime
EndTime
Sales
PK
SalesID
QuantitySold
SellingPrice
SaleDate
Product
PK
ProductID
ProductName
WholesaleCost
MinimumSellingPrice

Normalization Schema



Normalization From ER Diagram



VenueID	VenueName	StreetAddress	City	State	PostalCode	Description
V001	KingCircle	201 State Street	Chicago	Illinois	60656	NO instructions
V002	QueenCircle	301 Indiana Street	Chicago	Illinois	60626	NO instructions
V003	Harris Theatre	60 Adams Street	Chicago	Illinois	60616	Parking at 2nd Street

	EventID	VenueID	EventName	Startdate	Enddate	Description	Eventtype
▶	E001	V002	The Musical	2024-06-17	2024-06-20	Famous	Music Festival
	E002	V001	The Summer Beach	2024-05-04	2024-05-06	Beach	Waterfront Festival
	E003	V001	The Winter Beach	2024-12-03	2024-12-06	The Beach	Waterfront Festival

Physical Database Design

Design Base Relations

- Primary key uniquely identifies records
- Attribute domains specify data types
- Required attribute values specified in DDL
- General constraints also in DDL
- CREATE script defines SQL commands, tables, keys, and constraints.

Scripts for Database Design (Unfinished)

- We used NOT NULL for primary key attribute to reject any attempt to insert a null values in the column
- The Check in constraint helps us maintain consistency and accuracy in the database by preventing invalid or unexpected data from being entered into the Eventtype column.
- The FOREIGN KEY constraint establishes a relationship between the VenueID column in the current table and the VenueID column in another table named Venue.

```
CREATE TABLE Event(
  EventID VARCHAR(10) NOT NULL,
  VenueID VARCHAR(10) NOT NULL,
  EventName VARCHAR(50) NOT NULL,
  StartDate DATE NOT NULL,
  EndDate DATE NOT NULL,
  Description VARCHAR(255),
  EventType VARCHAR(25),
  CHECK (Eventtype IN ('Trade Show', 'Rib Fest',
'Waterfront festival', 'Music Festival', 'Beer Festival',
'Sporting Event', 'Street Festival')),
  PRIMARY KEY (EventID),
  FOREIGN KEY (VenueID) REFERENCES Venue(VenueID)
    ON UPDATE CASCADE
    ON DELETE NO ACTION);
```

Data Manipulation Language (DML) Statements (V)

- **INSERT: Insert new data into a table.**

```
INSERT INTO Venue (VenueID, VenueName,
StreetAddress, City, State, PostalCode, Description)
VALUES ('V001', 'Convention Center', '123 Main St',
Toronto, Ontario, '12345', 'Large venue for conventions
and trade shows');
```

- **UPDATE: Modify existing data in a table.**

```
UPDATE Event SET EventName = ‘FoxCore Summer
Festival’ WHERE EventID = ‘E002’;
```

- **DELETE: Remove data from a table.**

```
DELETE FROM Product WHERE ProductID = ‘P001’;
```

SQL Commands for Database

“The result from the SELECT statement provides the first name and last name of the salesperson, the event name, and the product name. It shows the total quantity of the product sold for each group combination of salesperson ID, event ID, and product ID. The results are grouped by salesperson ID, event ID, and product ID.”

```
Select sp.Firstname, sp.Lastname,e.eventname,
p.productname, SUM(s.quantitysold) AS Totalquantitysold
From sales s
JOIN Shift sh ON s.Shiftid=sh.shiftid
JOIN event e ON e.eventid=s.eventid
JOIN product p ON s.productid=p.productid
JOIN salesperson sp ON
sh.salespersonid=sp.salespersonid
Group by sp.salespersonid,e.eventid, p.productid;
```

Firstname	Lastname	eventname	productname	Totalquantitysold
John	Kite	The Musical	Bubble Guns	20
Ken	White	The Musical	Arctic Skin Cooling Towels	15
John	Kite	FoxComm Festival	Arctic Skin Cooling Towels	25
Lauren	Wolf	FoxComm Festival	Arctic Skin Cooling Towels	22

Thanks!