Exercise 2) Designing Entities

Senario1) A student at a university

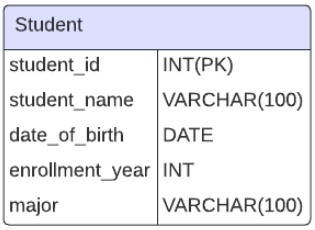
Step 1: Describe Each Table and Attributes

Table: Student

- student_id (Primary Key, INT)
- student_name (VARCHAR(100), Not Null)
- date_of_birth (DATE, Not Null)
- enrollment_year (YEAR, Not Null)
- major (VARCHAR(100))

- **student_id (Primary Key, INT):** Unique identifier for each student. This field serves as the primary key, operating on the INT data type. It uniquely identifies each student entry within the table.
- **student_name (VARCHAR(100), Not Null):** Represented as a VARCHAR data type, this attribute stores the name of the student, accommodating strings of variable lengths.
- **date_of_birth (DATE, Not Null):** Uses the DATE data type to store the birth date of the student.
- **enrollment_year (INT, Not Null):** This attribute stores the year the student enrolled. It employs the INT data type.
- **major (VARCHAR(100)):** This attribute stores the major field of study of the student, using the VARCHAR data type.

Step 2: Draw Database Schema with Relationship Single Table: This table is self-contained and does not have relationships with other tables in this context.



Step 3: Describe Each Association There are no associations to describe for this single table.

Senario2) A faculty member at a university.

Step 1: Describe Each Table and Attributes

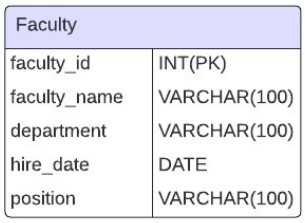
Table: Faculty

- faculty_id (Primary Key, INT, Auto Increment)
- faculty_name (VARCHAR(100), Not Null)
- department (VARCHAR(100), Not Null)
- hire_date (DATE, Not Null)
- position (VARCHAR(100), Not Null)

- **faculty_id (Primary Key, INT, Auto Increment):** Unique identifier for each faculty member. This field serves as the primary key, operating on the INT data type. It uniquely identifies each faculty member entry within the table.
- **faculty_name (VARCHAR(100), Not Null):** Represented as a VARCHAR data type, this attribute stores the name of the faculty member, accommodating strings of variable lengths.

• **department (VARCHAR(100), Not Null):** This attribute stores the name of the department the faculty member belongs to. It employs the VARCHAR data type.
• **hire_date (DATE, Not Null):** Uses the DATE data type to store the hire date of the faculty member.
• **position (VARCHAR(100), Not Null):** This attribute stores the position of the faculty member, using the VARCHAR data type.

Step 2: Draw Database Schema with Relationship Single Table: This table is self-contained and does not have relationships with other tables in this context.



Step 3: Describe Each Association There are no associations to describe for this single table.

Senario3) A work of art that is displayed in a gallery or museum.

Table: ArtWork

- artwork_id (Primary Key, INT)
- title (VARCHAR(100), Not Null)
- artist (VARCHAR(100), Not Null)
- year_created (INT)
- type (VARCHAR(50))
- gallery_id (Foreign Key, INT, References Gallery(gallery_id))

Table: Gallery

- gallery_id (Primary Key, INT)
- gallery_name (VARCHAR(100), Not Null)
- location (VARCHAR(100))

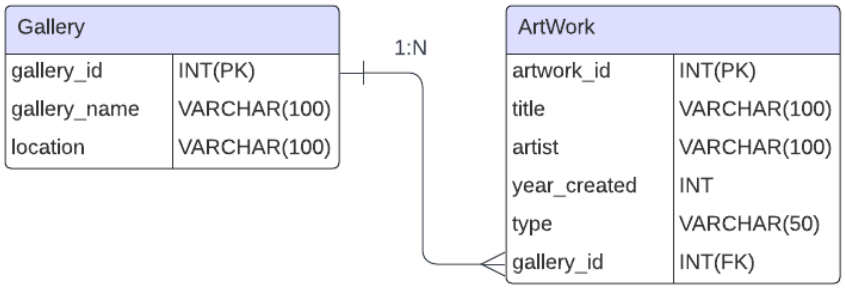
Step 1: Describe Each Table and Attributes

ArtWork Table
• **artwork_id (Primary Key, INT):** Unique identifier for each artwork. This field serves as the primary key, operating on the INT data type. It uniquely identifies each artwork entry within the table.
• **title (VARCHAR(100), Not Null):** Represented as a VARCHAR data type, this attribute stores the title of the artwork, accommodating strings of variable lengths.
• **artist (VARCHAR(100), Not Null):** This attribute stores the name of the artist. It employs the VARCHAR data type.
• **year_created (INT):** Uses the INT data type to store the year the artwork was created.
• **type (VARCHAR(50)):** This attribute stores the type of artwork, using the VARCHAR data type.
• **gallery_id (Foreign Key, INT, References Gallery(gallery_id)):** References the gallery_id in the Gallery table. This attribute establishes a foreign key relationship, indicating the gallery where the artwork is displayed.

Gallery Table
• **gallery_id (Primary Key, INT):** Unique identifier for each gallery. This field serves as the primary key, operating on the INT data type. It uniquely identifies each gallery entry within the table.
• **gallery_name (VARCHAR(100), Not Null):** Represented as a VARCHAR data type, this attribute stores the name of the gallery, accommodating strings of variable lengths.

• **location (VARCHAR(100)):** Uses the VARCHAR data type to store the location of the gallery.

Step 2: Draw Database Schema with Relationship



Step 3: Describe Each Association

Association from Gallery to ArtWork
• One-to-Many Relationship: Each gallery can have multiple artworks. This is represented by the gallery_id in the ArtWork table that references the gallery_id in the Gallery table.

Association from ArtWork to Gallery
• Many-to-One Relationship: Each artwork belongs to one gallery. This is represented by the foreign key gallery_id in the ArtWork table that links to the gallery_id in the Gallery table.

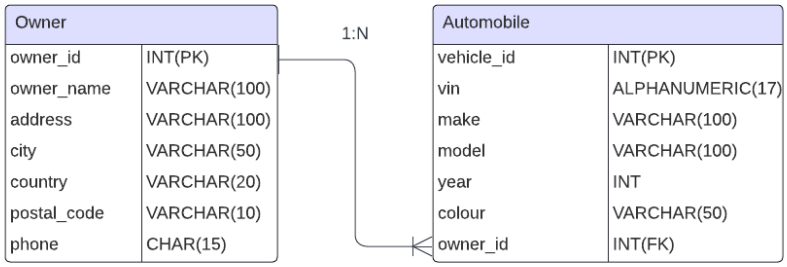
Senario4) An automobile that is registered with the Motor Vehicle Department.

Step 1: Describe Each Table and Attributes

Automobile Table
• **vin (Primary Key, CHAR(17)):** Unique identifier for each automobile.
This field serves as the primary key, operating on the CHAR data type with a fixed length of 17. It uniquely identifies each automobile entry within the table.
• **make (VARCHAR(50), Not Null):** Represented as a VARCHAR data type, this attribute stores the make of the automobile, accommodating strings of variable lengths.
• **model (VARCHAR(50), Not Null):** This attribute stores the model of the automobile. It employs the VARCHAR data type.
• **year (INT, Not Null):** Uses the INT data type to store the year the automobile was manufactured.
• **color (VARCHAR(50), Not Null):** This attribute stores the color of the automobile, using the VARCHAR data type.
• **owner_id (Foreign Key, INT, References Owner(owner_id)):** References the owner_id in the Owner table. This attribute establishes a foreign key relationship, indicating the owner of the automobile.

Owner Table
• **owner_id (Primary Key, INT):** Unique identifier for each owner. This field serves as the primary key, operating on the INT data type. It uniquely identifies each owner entry within the table.
• **owner_name (VARCHAR(100), Not Null):** Represented as a VARCHAR data type, this attribute stores the name of the owner, accommodating strings of variable lengths.
• **address (VARCHAR(100)):** Uses the VARCHAR data type to store the address of the owner.
• **city (VARCHAR(50)):** This attribute stores the city where the owner resides, using the VARCHAR data type.
• **country (VARCHAR(20)):** Uses the VARCHAR data type to store the country of the owner.
• **postal_code (VARCHAR(10)):** This attribute stores the postal code of the owner's address, using the VARCHAR data type.
• **phone (VARCHAR(15)):** Uses the VARCHAR data type to store the phone number of the owner.

Step 2: Draw Database Schema with Relationship



Step 3: Describe Each Association

Association from Owner to Automobile
• One-to-Many Relationship: Each owner can have multiple automobiles. This is represented by the owner_id in the Automobile table that references the owner_id in the Owner table.

Association from Automobile to Owner
• Many-to-One Relationship: Each automobile belongs to one owner. This is represented by the foreign key owner_id in the Automobile table that links to the owner_id in the Owner table.

Senario5) Pizza

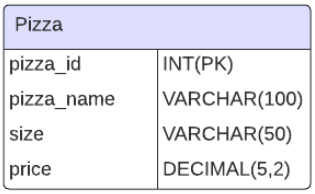
Step 1: Describe Each Table and Attributes

Table: Pizza

- pizza_id (Primary Key, INT)
- pizza_name (VARCHAR(100), Not Null)
- size (VARCHAR(50), Not Null)
- price (DECIMAL(5, 2), Not Null)

• **pizza_id (Primary Key, INT, Auto Increment):** Unique identifier for each pizza. This field serves as the primary key, operating on the INT data type. It uniquely identifies each pizza entry within the table.
• **pizza_name (VARCHAR(100), Not Null):** Represented as a VARCHAR data type, this attribute stores the name of the pizza, accommodating strings of variable lengths.
• **size (VARCHAR(50), Not Null):** This attribute stores the size of the pizza. It employs the VARCHAR data type.
• **price (DECIMAL(5, 2), Not Null):** Uses the DECIMAL data type with a precision of 5 and scale of 2 to store the price of the pizza.

Step 2: Draw Database Schema with Relationship



Step 3: Describe Each Association

There are no associations to describe for this single table.