

# Conceptual Database Design

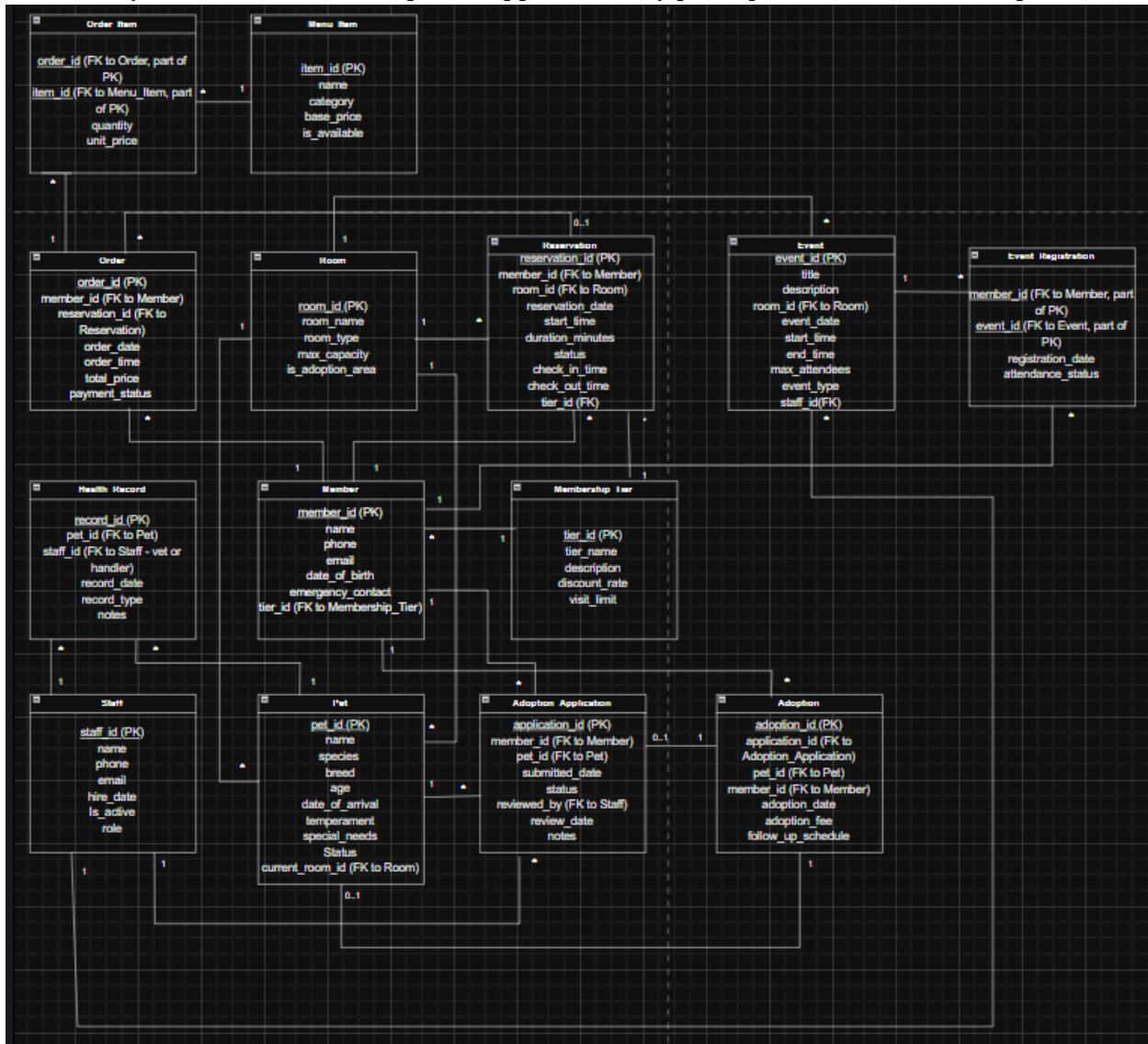
## 1. Historical Membership Tier

Although each Member has a current membership tier, the Reservation entity stores the tier at the time of the visit. This ensures accurate reporting when membership changes over time (Query 2).

## 2. Adoption Rules

Every Adoption must reference exactly one Adoption\_Application and exactly one Pet.

A Pet or Application may exist without an adoption, but an Adoption cannot exist without both. Additionally, each Pet and each Adoption\_Application may participate in at most one Adoption.



## Logical Database Design

After we completed our ER Diagram, we had to convert it into actual tables in Oracle's SQL. The schema of our tables can be seen in the tables below:

**Membership\_Tier**

<i>tier_id</i>	INTEGER
<i>tier_name</i>	VARCHAR(50) NOT NULL
<i>description</i>	VARCHAR(200)
<i>discount_rate</i>	NUMBER(3) (BETWEEN 0 AND 100)
<i>visit_limit</i>	INTEGER

**Room**

<i>room_id</i>	INTEGER
<i>room_name</i>	VARCHAR(100) NOT NULL
<i>room_type</i>	VARCHAR(50)
<i>max_capacity</i>	INTEGER NOT NULL
<i>is_adoption_area</i>	NUMBER(1) (Must be 0 or 1)

**Member**

<i>member_id</i>	INTEGER
<i>name</i>	VARCHAR2(100) NOT NULL
<i>phone</i>	VARCHAR(20)
<i>email</i>	VARCHAR2(100)
<i>date_of_birth</i>	DATE
<i>emergency_contact</i>	VARCHAR2(200)
<i>tier_id</i>	INTEGER (FK to Membership_Tier)

**Reservation**

<i>reservation_id</i>	INTEGER
<i>member_id</i>	INTEGER NOT NULL (FK to Member)
<i>room_id</i>	INTEGER NOT NULL (FK to Room)
<i>start_time</i>	DATE NOT NULL
<i>duration_minutes</i>	INTEGER NOT NULL
<i>status</i>	VARCHAR(20)
<i>check_in_time</i>	TIMESTAMP NOT NULL
<i>check_out_time</i>	TIMESTAMP NOT NULL
<i>tier_id</i>	INTEGER (FK to Membership_Tier)

**Staff**

<i>staff_id</i>	INTEGER
name	VARCHAR2(100) NOT NULL
phone	VARCHAR(20)
email	VARCHAR2(100)
hire_date	DATE
is_active	NUMBER(1) (Must be 0 or 1)
role	VARCHAR2(50)

**Health\_Record**

<i>record_id</i>	INTEGER
<i>pet_id</i>	(FK to Pet)
<i>staff_id</i>	INTEGER NOT NULL (FK to Staff)
<i>record_date</i>	DATE NOT NULL
<i>record_type</i>	VARCHAR2(50)
notes	VARCHAR2(500)

**Pet**

<i>pet_id</i>	INTEGER
name	VARCHAR2(100) NOT NULL
species	VARCHAR2(100)
breed	VARCHAR2(100)
age	INTEGER
date_of_arrival	DATE
temperament	VARCHAR2(200)
special_needs	VARCHAR2(200)
status	VARCHAR2(30)
<i>current_room_id</i>	INTEGER (FK to Room)

**Adoption\_Application**

<i>application_id</i>	INTEGER
<i>member_id</i>	INTEGER NOT NULL (FK to Member)
<i>pet_id</i>	INTEGER NOT NULL (FK to Pet)
<i>submitted_date</i>	DATE NOT NULL
<i>status</i>	VARCHAR2(20)
<i>reviewed_by</i>	INTEGER (FK to Staff)
<i>review_date</i>	DATE
notes	VARCHAR(500)

### Adoption

<i>adoption_id</i>	INTEGER
<i>application_id</i>	INTEGER NOT NULL (FK to Adoption_Application)
<i>pet_id</i>	INTEGER NOT NULL (FK to Pet)
<i>member_id</i>	INTEGER NOT NULL (FK to Member)
<i>adoption_date</i>	DATE
<i>adoption_fee</i>	NUMBER(7, 2)
<i>follow_up_schedule</i>	DATE

### Customer\_Order

<i>order_id</i>	INTEGER
<i>member_id</i>	INTEGER (FK to Member)
<i>reservation_id</i>	INTEGER (FK to Reservation)
<i>order_date</i>	DATE
<i>order_time</i>	TIMESTAMP
<i>total_price</i>	NUMBER(8, 2)
<i>payment_status</i>	VARCHAR2(20)

### Menu\_Item

<i>item_id</i>	INTEGER
<i>name</i>	VARCHAR(100) NOT NULL
<i>category</i>	VARCHAR(50)
<i>base_price</i>	NUMBER(6, 2) NOT NULL
<i>is_available</i>	NUMBER(1) (Must be 0 or 1)

### Order\_Item

<i>order_id</i>	INTEGER (FK to Customer_Order)
<i>item_id</i>	INTEGER (FK to Menu_Item)
<i>quantity</i>	INTEGER
<i>unit_price</i>	NUMBER(7,2) NOT NULL

**Event**

<i>event_id</i>	INTEGER
<i>title</i>	VARCHAR(100)
<i>description</i>	VARCHAR(400)
<i>room_id</i>	INTEGER NOT NULL (FK to Room)
<i>event_date</i>	DATE NOT NULL
<i>start_time</i>	TIMESTAMP
<i>end_time</i>	TIMESTAMP
<i>max_attendees</i>	INTEGER NOT NULL
<i>event_type</i>	VARCHAR2(50)
<i>staff_id</i>	INTEGER NOT NULL (FK to Staff)

**Event\_Registration**

<i>member_id</i>	INTEGER (FK to Member)
<i>event_id</i>	INTEGER (FK to Event)
<i>registration_date</i>	DATE
<i>attendance_status</i>	VARCHAR(30)

## Normalization Analysis

All of the FDs are as follows:

- Order\_Item(order\_id, item\_id) → quantity, unit\_price
- Menu\_Item(item\_id) → name, category, base\_price, is\_available
- Customer\_Order(order\_id) → member\_id, reservation\_id, order\_date, order\_time, total\_price, payment\_status
- Room(room\_id) → room\_name, room\_type, max\_capacity, is\_adoption\_area
- Reservation(reservation\_id) → member\_id, room\_id, reservation\_date, start\_time, duration\_minutes, status, check\_in\_time, check\_out\_time, tier\_id
- Event(event\_id) → title, description, room\_id, event\_date, start\_time, end\_time, max\_attendees
- Event\_Registration(member\_id, event\_id) → registration\_date, attendance\_status
- Health\_Record(record\_id) → pet\_id, staff\_id, record\_date, record\_type, notes
- Membership\_Tier(tier\_id) → tier\_name, description, discount\_rate, visit\_limit
- Staff(staff\_id) → name, phone, email, hire\_date, is\_active, role
- Member(member\_id) → name, phone, email, date\_of\_birth, emergency\_contact, tier\_id
- Pet(pet\_id) → name, species, breed, age, date\_of\_arrival, temperament, special\_needs, status, current\_room\_id
- Adoption\_Application(application\_id) → member\_id, pet\_id, submitted\_date, status, reviewed\_by, review\_date, notes
- Adoption(adoptions\_id) → application\_id, pet\_id, member\_id, adoption\_date, adoption\_fee, follow\_up\_schedule

In order to know if our tables are in 3NF / BCNF, we must first look at the definitions of 1NF and 2NF. First of all, the tables are clearly in 1NF form because none of our data is in set form, meaning each entry under a column only contains the one value. We also know that the tables are in 2NF because of our two tables that have multiple candidate keys Order\_Item and Event\_Registration.

We know that Order\_Item relies on both the order\_id and item\_id because you need both items to be able to uniquely identify an order since an item can show up in multiple orders and an order can have multiple items. We also need both the member\_id and event\_id to uniquely identify an event registration since a member can go to multiple events and an event has multiple members that can attend.

Lastly, to ensure we are in 3NF / BCNF we know that none of our tables have transitive dependencies. All of our non-primary key attributes only depend on primary keys.

## **Query Description**

### **Query 1 - Adoption Applications for a Pet**

This query lists all adoption applications submitted for a specific pet.

It joins Adoption\_Application, Member, and Staff to show:

- Applicant name
- Application date
- Current status
- Assigned coordinator

This helps staff track interest and monitor adoption progress.

### **Query 2 - Visit History for a Member**

This query provides a complete history of a member's visits and spending.

It joins Reservation, Room, Membership\_Tier, and Customer\_Order to display:

- Reservation date and time
- Room visited
- Membership tier at the time
- Total spending for each visit

This supports customer analysis and billing review.

### **Query 3 - Upcoming Events with Available Capacity**

This query identifies upcoming events that are not yet full.

It combines Event, Event\_Registration, Room, and Staff to report:

- Event name, date, and time
- Room location
- Current number of attendees vs. capacity
- Event coordinator

Useful for planning and promoting events.

### **Query 4 - Custom Query: Top Members by Spending**

This query ranks high-value members based on total spending.

It joins Member, Membership\_Tier, Reservation, and Customer\_Order, and filters by a user-entered minimum spending amount.

The output includes:

- Member ID and name
- Tier
- Number of visits
- Total spending

This helps management understand customer value and behavior.