IRCTC

### 1. IRCTC

Entities : Train, Ticket, Customer

Tickets can be reserved or unreserved.

Each reserved ticket must correspond to a train and a customer.

Each unreserved ticket must correspond to a customer.

A customer cannot be in two trains at the same time.

#### 1. Train (Strong Entity)

* Each train has a unique train number.
* Attributes: train\_id, train\_name, source, destination, departure\_time, arrival\_time.

#### 2. Customer (Strong Entity)

* Each customer has a unique customer ID.
* Attributes: customer\_id, name, email, phone.

#### 3. Ticket (Weak Entity)

* Ticket depends on Customer (and optionally Train if reserved).
* Attributes: ticket\_id, type (reserved/unreserved), booking\_date, seat\_number (nullable if unreserved).
* Relationship:  
  + Reserved Ticket ⟶ linked to both Customer and Train
  + Unreserved Ticket ⟶ linked only to Customer

## 

## 

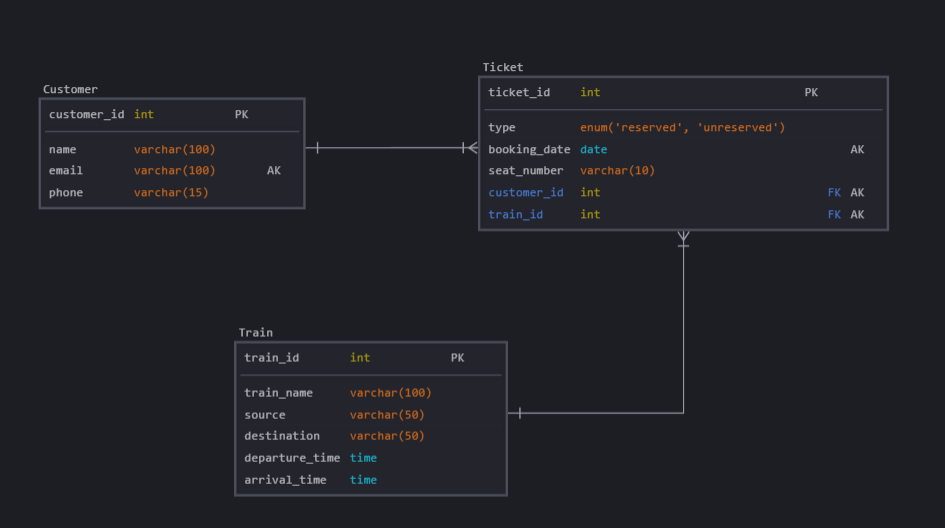
#### Relationships

#### 1. Books (Customer ↔ Ticket)

* One customer can book many tickets.
* Each ticket belongs to exactly one customer.

#### 2. Assigned\_To (Ticket ↔ Train)

* Only reserved tickets are assigned to trains.
* Unreserved tickets do not need this relationship.



Hotel Booking

### 2. Hotel Booking

Entities : Hotel, Customer

Customers can be first time users or loyal customers.

Customers can book from only one hotel at a time. A hotel can have many customers.

#### 1. Hotel – *Strong Entity*

* Each hotel is uniquely identified.
* Attributes: inn\_id, inn\_name, location, rating.

#### 2. Guest (for Customer) – *Strong Entity*

* Each customer is uniquely identified.
* Type: first-time user / loyal customer.
* Attributes: guest\_id, name, email, phone, guest\_type.

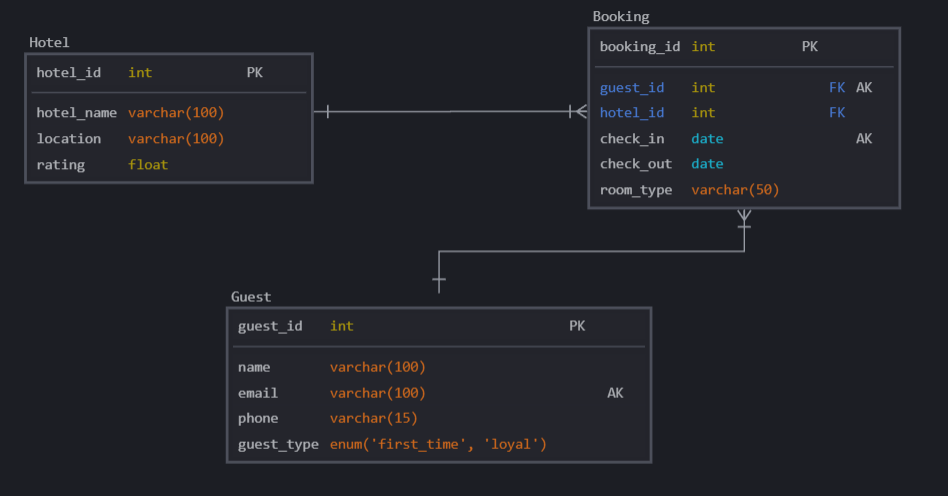
#### 3. Booking – *Weak Entity*

* Represents a booking done by a guest at a specific inn.
* A guest can have only one active booking at a time.
* Attributes: booking\_id, guest\_id, inn\_id, check\_in, check\_out, room\_type.

#### Relationships

* A **Guest** can make multiple **Bookings**, but **only one active booking** at a time.
* A **Booking** must be associated with exactly **one Inn**.
* An **Inn** can serve multiple **Guests**.

#### Hotel\_Booking ER- Diagram



E-Commerce

### 3. E-Commerce

Entities : Supplier, Customer, Items, Order

Every item should correspond to a supplier.

One supplier can have more than one item.

A customer can have one order at the same time.

One order can have multiple items from multiple brands.

#### 1. Vendor – *Strong Entity*

* Supplies products sold on the platform.
* Attributes: vendor\_id, vendor\_name, contact\_email, location.

#### 2. Shopper – *Strong Entity*

* Make purchases on the app.
* Attributes: shopper\_id, name, email, phone.

#### 3. Product – *Strong Entity*

* Must be associated with one Vendor.
* Attributes: product\_id, product\_name, price, stock, category, vendor\_id.

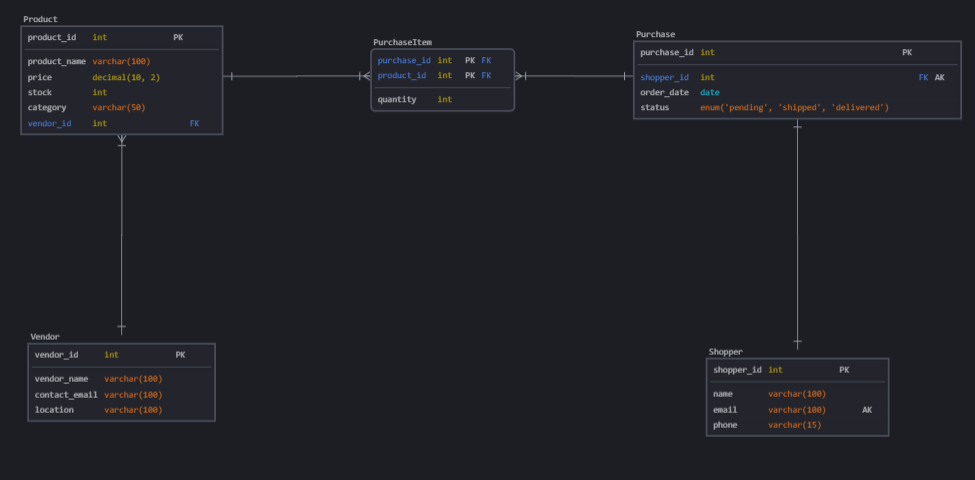
#### 4. Purchase – *Weak Entity*

* Represents an order made by a Shopper.
* A shopper can only have one active order at a time.
* Attributes: purchase\_id, shopper\_id, order\_date, status.

#### 5. PurchaseItem (Weak Entity)

* Captures multiple items per order (many-to-many between Purchase and Product).
* Attributes: purchase\_id, product\_id, quantity.

#### E-Commerce E-R Diagram



IMBD

### 4. IMBD

Entities : Movie, Actors, TV series

A TV series or a movie must have an actor. An actor can act in both.

#### 1. Actor – *Strong Entity*

* Attributes: actor\_id, name, birthdate, gender, nationality.

#### 2. Movie – *Strong Entity*

* Attributes: movie\_id, title, release\_year, genre, duration.

#### 3. TVSeries – *Strong Entity*

* Attributes: series\_id, title, start\_year, end\_year, genre, num\_seasons.

#### 

#### 4. ActsInMovie – *Weak Entity*

* Links actors to movies (many-to-many).
* Attributes: actor\_id, movie\_id, role.

#### 5. ActsInSeries – *Associative Entity*

* Links actors to TV series (many-to-many).
* Attributes: actor\_id, series\_id, role.

#### IMBD ER-Diagram



Paytm

### 5. Paytm

Entities : Accounts, Customer, Branches.

Each customer must have an account. Joint accounts are allowed.

A customer can have multiple accounts in different branches, but not in the same branch.

A branch can have many accounts.

#### 1. Customer – *Strong Entity*

* A person who owns one or more accounts.
* Attributes: customer\_id, name, email, phone, address.

#### 2. Branch – *Strong Entity*

* A physical bank location.
* Attributes: branch\_id, branch\_name, location, ifsc\_code.

#### 3. Account – *Strong Entity*

* A financial account held by one or more customers at a branch.
* Attributes: account\_id, account\_type, balance, branch\_id.

#### 4. AccountHolder – *Weak Entity*

* Links customers to accounts (for joint accounts).
* Attributes: account\_id, customer\_id.

#### 

#### Relationship

1. **Each customer must have at least one account.**
2. **A customer can have multiple accounts**, but **not in the same branch**.  
    → Enforced with a **unique constraint** on (customer\_id, branch\_id) via a derived relationship.
3. **Each branch can have many accounts.**

#### Paytm ER-Diagram

