Data Modelling

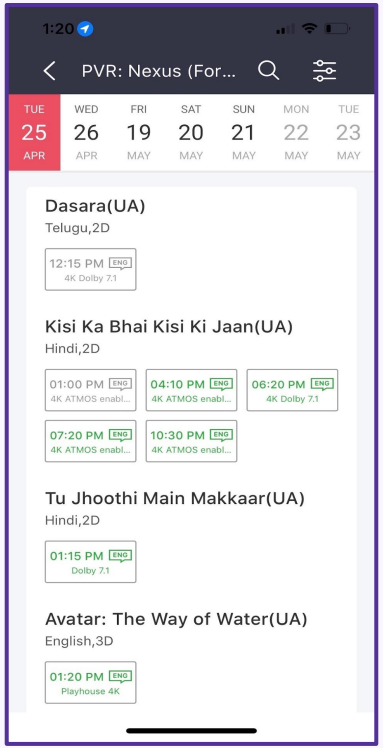
Sancheeta Kaushal | Ex-EM Blinkit

Problem Statement

Problem Solving Case: Bookmyshow

Bookmyshow is a ticketing platform where you can book tickets for a movie show. As part of this assignment, we need to build API's for the following feature. As a user, I can select any theatre in the city. On selecting the theatre, I should be able to see the dates of next 7 days. I can click on any date and the page should load to give me all the movies in that theatre on that given date. Movies should contain details of all the showtimes.

Problem Statement



The image represents the feature described on the next page. You have to code the APIs along with appropriate table structures for powering this UI.

Airtribe

What all skills you need to learn to build this ?

- Storing data in a table.
- Accessing data using SQL Queries.
- Normalizing data for efficient storage.
- Joining data across multiple tables.
- Understanding the role of transactions.
- Solving for the database challenges
- Accessing data from the tables in the application layer using ORMs.

What all skills you need to learn to build this ?

- Storing data in a table.
- Accessing data using SQL Queries.
- Normalizing data for efficient storage.
- Joining data across multiple tables.
- Understanding the role of transactions.
- Solving for the database challenges
- Accessing data from the tables in the application layer using ORMs.

Agenda

- What and why of data modelling ?
- A brief history of databases
- Concept of entity, entity type and attributes
- Intro to SQL.
- Database Normalisation - 1NF, 2NF, BCNF

What is data modelling?

- Modelling complex real world relationships in software domain

  Like a Building Plan from an Architect

Why is data modelling required?

- Organises complex business and science data like Apollo, Airlines
- Improves database performance by reducing the data to be queried
- Enables communication by defining the data relationships
- Reduces chances of data inconsistencies and errors

Why should you learn data modelling ?

- What do you think a day for a backend engineer looks like ?
- A lot of downtimes in companies can be attributed to changes in databases.
- Lastly, job interviews.

A brief history

1. File systems
2. IDS, IMS & CODASYL
3. RDBMS
4. NoSQL

Fundamentals of DBMS

- Entity Value
  - Object or thing in a real world
- Entity Type
  - Collection of entities
- Attribute
  - Property/Behaviour of an entity

Example- Entity & Attributes

One can buy 1 kg onion at Rs. 12 and 1kg potato at Rs. 10. We have 20 kg onion and 30 kg potato available at the Instamart store.

Define the entity, entity type, attributes in this given scenario ?

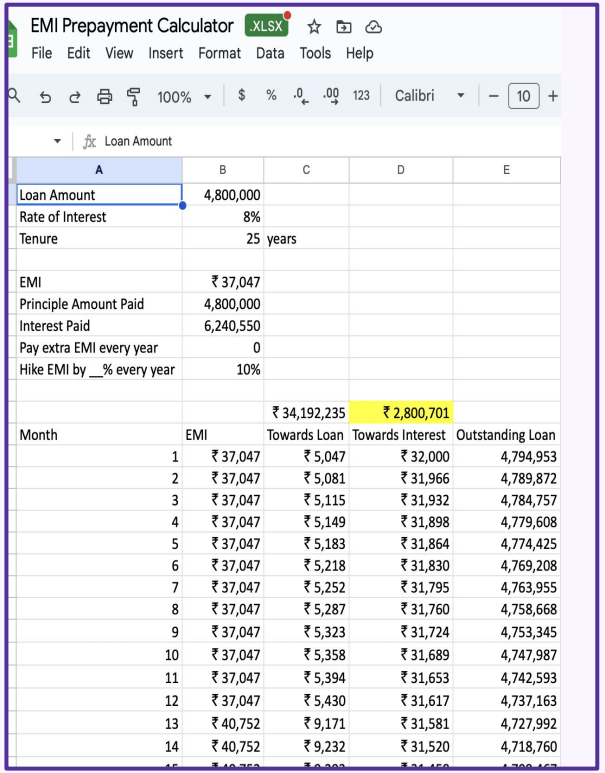
Example– Entity & Attributes

One can buy 1 kg onion at Rs. 12 and 1kg potato at Rs. 10. We have 20 kg onion and 30 kg potato available at the Instamart store.

Entity Type - Products
Entity Value - Onion and Potato
Attributes - Price and Stock

Fundamentals of DBMS

- Relational Database
  - Structuring entity type and values and their attributes in a table form.
- Tables
  - Data structure with rows and columns.



EMI Prepayment Calculator .XLSX

File Edit View Insert Format Data Tools Help

Loan Amount

	A	B	C	D	E
1	Loan Amount	4,800,000			
2	Rate of Interest	8%			
3	Tenure	25	years		
4					
5	EMI	₹ 37,047			
6	Principle Amount Paid	4,800,000			
7	Interest Paid	6,240,550			
8	Pay extra EMI every year	0			
9	Hike EMI by __% every year	10%			
10					
11			₹ 34,192,235	₹ 2,800,701	
12	Month	EMI	Towards Loan	Towards Interest	Outstanding Loan
13	1	₹ 37,047	₹ 5,047	₹ 32,000	4,794,953
14	2	₹ 37,047	₹ 5,081	₹ 31,966	4,789,872
15	3	₹ 37,047	₹ 5,115	₹ 31,932	4,784,757
16	4	₹ 37,047	₹ 5,149	₹ 31,898	4,779,608
17	5	₹ 37,047	₹ 5,183	₹ 31,864	4,774,425
18	6	₹ 37,047	₹ 5,218	₹ 31,830	4,769,208
19	7	₹ 37,047	₹ 5,252	₹ 31,795	4,763,955
20	8	₹ 37,047	₹ 5,287	₹ 31,760	4,758,668
21	9	₹ 37,047	₹ 5,323	₹ 31,724	4,753,345
22	10	₹ 37,047	₹ 5,358	₹ 31,689	4,747,987
23	11	₹ 37,047	₹ 5,394	₹ 31,653	4,742,593
24	12	₹ 37,047	₹ 5,430	₹ 31,617	4,737,163
25	13	₹ 40,752	₹ 9,171	₹ 31,581	4,727,992
26	14	₹ 40,752	₹ 9,232	₹ 31,520	4,718,760

Airtribe

Example- Define the table

One can buy 1 kg onion at Rs. 12 and 1kg potato at Rs. 10. We have 20 kg onion and 30 kg potato available at the Instamart store.

Represent the data in form of a table and model the entities described in the above scenario ? What would be the rows and the columns ?

Example- Define the table

One can buy 1 kg onion at Rs. 12 and 1kg potato at Rs. 10. We have 20 kg onion and 30 kg potato available at the Instamart store.

Table to model the entities above

Table name - product

Id	Name	Price (in Rs.)	Stock (in kg)
1	Onion	12	20
2	Potato	10	30

Intro to SQL

- SQL as DSL
    - Structured Query Language as Domain Specific Language that helps us to get data from tables. Focus on what to do not how to do unlike API's.
- Data Types
    - Varchar, Int, Bigint, Text, Float, DateTime, Decimal, JSON etc.
- SQL Clauses
    - Function that helps with filtering and manipulation of data. Eg. WHERE, HAVING, GROUP BY, ORDER BY, LIKE etc.

Concept of Keys & Constraints

- **Primary Key**
  - Key used to uniquely identify a row in a table. Eg. user id in user table
- **Foreign Key**
  - Primary key of one table being used as a key in another table. Eg. user id in order table
- **Candidate Key**
  - Multiple keys being used to uniquely identify a row in a table. Eg. Phone + Customer ID
- **Constraints**
  - Rules for data in a table. Eg. NOT NULL, UNIQUE, DEFAULT, PRIMARY KEY, FOREIGN KEY

SQL Operations

- **DDL – Data Definition & Language**
  - Create, alter, truncate and drop in a table
- **DML – Data Manipulation Language**
  - Insert, update and delete in a table.
- **DCL – Data Control Language**
  - Grant and revoke access on database, table and column.
- **DQL – Data Query Language**
  - Select data in a table.

Create database & Insert table syntax

SQL Query

Syntax for creating a table

CREATE TABLE table name (column datatype NOT NULL AUTO_INCREMENT, column1 datatype, column2 datatype, ... columnN datatype, PRIMARY KEY (column1));

Syntax for inserting a new row to a table

INSERT INTO table_name (column1, column2, ... columnN) VALUES (value1, value2, valueN);

Syntax for selecting data from table

SELECT column1, column2, ... columnN from table_name;

Example – Create table

One can buy 1 kg onion at Rs. 12 and 1kg potato at Rs. 10. We have 20 kg onion and 30 kg potato available at the Instamart store.

Write the SQL Query to create this table.

Table name - Product

Id	Name	Price (in Rs.)	Stock (in kg)
1	Onion	12	20
2	Potato	10	30

Example – Create table

SQL Query

```
CREATE TABLE product (id int NOT NULL AUTO_INCREMENT, name varchar(120), price int, stock float, PRIMARY KEY (id));
INSERT INTO product (name, price, stock) VALUES ('Onion', 12, 20);
INSERT INTO product (name, price, stock) VALUES ('Potato', 10, 30);
```

Id	Name	Price (in Rs.)	Stock (in kg)
1	Onion	12	20
2	Potato	10	30

Example – Add new row to table

Instamart has decided to **sell Tomato at price per kg as Rs. 25**.
They have purchased a stock of 40 kg.

Write the SQL Query to add this new information to the table.

Table name - Product

Id	Name	Price (in Rs.)	Stock (in kg)
1	Onion	12	20
2	Potato	10	30

Example – Add new row to table

SQL Query

```
INSERT INTO product (name, price, stock) VALUES (‘Tomato’, 25, 40);
```

Id	Name	Price (in Rs.)	Stock (in kg)
1	Onion	12	20
2	Potato	10	30
3	Tomato	25	40

Example – Update existing attribute

As part of the daily milk run, additional 30 kg onion was bought.

Represent the updated information in the product table.

Table name - Product

Id	Name	Price (in Rs.)	Stock (in kg)
1	Onion	12	20
2	Potato	10	30
3	Tomato	25	40

Example – Update existing row syntax

SQL Query

```
UPDATE  table_name SET updateColumn = UpdateValue WHERE
uniqueColumn= uniqueValue;
```

Example – Update existing attribute

SQL Query

```
UPDATE product SET stock = 50 WHERE id = 1;
```

Id	Name	Price (in Rs.)	Stock (in kg)
1	**Onion**	**12**	**50**
2	Potato	10	30
3	Tomato	25	40

Example – Remove a product

Instamart has decided to stop selling potato because the quality is bad.

Remove potato from the following table.

Table name - Product

Id	Name	Price (in Rs.)	Stock (in kg)
1	Onion	12	20
2	Potato	10	30
3	Tomato	25	40

Example – Remove row syntax

SQL Query

```
DELETE FROM table_name where  uniqueColumn =uniqueValue ;
```

Alternate way used in real world.

```
ALTER TABLE table_name add column column1 datatype DEFAULT value1;
UPDATE table_name set column2=value2 where column1 = value1;
```

Example – Remove a product

SQL Query

```
DELETE FROM product where id = 2; OR
ALTER TABLE product add column enabled bool default 1;
UPDATE product set enabled=0 where id = 2;
```

Id	Name	Price (in Rs.)	Stock (in kg)	Enabled
1	Onion	12	50	1
2	**Potato**	**10**	**30**	**0**
3	Tomato	25	40	1

Quick Activity – 5 mins

You are an Airtribe student undergoing Backend Engineering Course. Write SQL for the following scenarios.

Task 1: Create table for storing your profile data in a table with attributes - Student name, Learning minutes, Attendence and Designation.
Task 2: Insert data for student named Sancheeta. Her learning minutes are 120 mins, attendance is 4 lectures, designation is Software Engineer.
Task 3: Update table to change her attendance to 5.

Data Aggregation and Ordering

- Group by and Having clause
- Aggregate functions like COUNT, SUM, AVG, MAX, or MIN
- Having vs Where - Full table scan
- Order by - Ascending or descending

Example – Group by

At Instamart, you have orders table and you want to find out how much revenue is generated by each customer.

SQL Query Syntax

```
SELECT Column1, aggregate_function(Column2)
FROM TableA
GROUP BY Column1;
```

SQL Query Syntax

```
SELECT customer_id, SUM(total_amount) as revenue
FROM orders
GROUP BY customer_id;
```

Example – Having

You have orders table and you want to find out how much revenue is generated by sales of each product and select only the products whose revenue is greater than 1050.

SQL Query Syntax

```
SELECT Column1, aggregate_function(Column2) as Column
FROM TableA
GROUP BY Column1 HAVING Column > X;
```

SQL Query Syntax

```
SELECT product_id, SUM(total_amount) as revenue
FROM orders
GROUP BY product_id HAVING revenue > 1050;
```

Example – Order by

Order the output of previous example in the descending order of revenue.

SQL Query Syntax

```
SELECT *
FROM Table
ORDER BY Column ASC/DESC;
```

SQL Query Syntax

```
SELECT customer_id, SUM(total_amount) as revenue
FROM orders
GROUP BY customer_id HAVING revenue > 1050 order by revenue desc;
```

Example – Grouping and Ordering

Given the data for orders with city, answer the following question

- How many orders were placed by each city, ordered in ascending order of the city name?

SQL Query Syntax

```
SELECT o.city,  COUNT(o.order_id) AS order_count
FROM orders o
GROUP BY o.city
ORDER BY o.city ASC;
```

Airtribe

Questions so far ?

Airtribe

How do we decide when to create a new table or when to create a new column in an existing table ?

- Normalization
- Think about when something transitions from an attribute to an entity

Normalisation

- Process of organising data in a database where you group similar values as one.
- Relational Algebra by E.F.Codd.
- Eliminate redundancy and inconsistent dependency.
- 1NF, 2NF, 3NF, BCNF

Normal forms

- **1NF**
  - Eliminate repeating groups in individual tables.
  - Ensure there is no multi-valued attribute.
  - Identify each set of related data with a primary key.

Example – 1NF

Instamart stores customer data to maintain their order history. Keep in mind to eliminate repeating groups.

Convert the following data into 1NF

Id	Name	Phone
1	Sancheeta	7272826385, 9064738238
2	Saurav	8574783832
3	Dhaval	7390372389, 8589830302

Example – 1NF

Showcasing the data in 1NF

Cust_Id	Name	Phone
1	Sancheeta	7272826385
1	Sancheeta	9064738238
2	Saurav	8574783832
3	Dhaval	7390372389
3	Dhaval	8589830302

Normal forms

- **2NF**
  - Create separate tables for sets of values that apply to multiple records.
  - Relate these tables with a foreign key.

Example – 2NF

Instamart stores customer data to maintain their order history. Keep in mind to eliminate partial functional dependence on the primary key.

Convert the following data into 2NF

Cust_Id	Name	Phone
1	Sancheeta	7272826385
1	Sancheeta	9064738238
2	Saurav	8574783832
3	Dhaval	7390372389
3	Dhaval	8589830302

Example – 2NF

Converting previous data into 2NF

Customer Details

Id	Name
1	Sancheeta
2	Saurav
3	Dhaval

Customer Phone Mapping

Id	Customer_Id	Phone
1	1	7272826385
2	2	9064738238
3	2	8574783832
4	3	7390372389
5	3	8589830302

Normal forms

- **3NF**
  - If an attribute depends on another non-key attribute rather than the primary key, it should be moved to a separate table.
  - Transitive dependency shouldn’t exist in same table.
    - A → B and B → C then A → C given B !→ A.
    - A: Employee ID, B: Department ID, C: Department Name
    - A → B (Employee ID determines Department ID)
    - B → C (Department ID determines Department Name)

Example – 3NF

Instamart recently started storing customer location details. Keep in mind to eliminate transitive dependency.

Showcase the following data in 3NF:

Id	Name	State	City	Pincode
1	Sancheeta	HP	Shimla	170021
2	Saurav	Karnataka	Bengaluru	560068
3	Dhaval	Karnataka	Coorg	571201

Example – 3NF

Showcasing the data in 3NF

Customer Details

Id	Name
1	Sancheeta
2	Saurav
3	Dhaval

Location details

Id	Customer_Id	State	City	Pincode
1	1	HP	Shimla	170021
2	2	Karnataka	Bengaluru	560068
3	3	Karnataka	Coorg	571201

Normal forms

- **BCNF**
  - Eliminate functional dependencies of attributes on anything other than a superset of a candidate key.
  - Keep in mind, we may not always normalize data in this form keeping in mind performance considerations. The more you normalize the more you need to join later.

Example – BCNF

Saurav went to his friends house and placed an order from there leading to multiple pincode. Which is the candidate key that can help you remove redundancy ? Keep in mind to eliminate multi-value dependency.

Showcase the following data in BCNF

Cust_Id	Name	State	City	Pincode
1	Sancheeta	HP	Shimla	170021
2	Saurav	Karnataka	Bengaluru	560093
2	Saurav	Karnataka	Bengaluru	560068
3	Dhaval	Karnataka	Coorg	571201

Example – BCNF

Showcasing the data in BCNF

Customer Details

Id	Name
1	Sancheeta
2	Saurav
3	Dhaval

Location details

Id	Cust_Id	State	City
1	1	HP	Shimla
2	2	Karnataka	Bengaluru
3	3	Karnataka	Coorg

City Pincode Mapping

Location_Id	Pincode
1	170021
2	560093
2	560068
3	571201

Example

Normalise the following data – https://bit.ly/41q4brG

Id	Student	Batch	Course	Teacher	Grade
1	Sancheeta	CSE	AI	Prof. Ajay	B
2	Navneet	CSE	AI	Prof. Ajay	A
2	Navneet	CSE	Database	Prof. Aman	B
3	Kartikay	ECE	OS	Prof. Deep	A
3	Kartikay	ECE	Database	Prof. Aman	B
4	Samyak	IT	OS	Prof. Deep	B

1NF

Id	Student	Batch	Course	Teacher	Grade
1	Sancheeta	CSE	AI	Prof. Ajay	B
2	Navneet	CSE	AI	Prof. Ajay	A
2	Navneet	CSE	Database	Prof. Aman	B
3	Kartikay	ECE	OS	Prof. Deep	A
3	Kartikay	ECE	Database	Prof. Aman	B
4	Samyak	IT	OS	Prof. Deep	B

2NF

Id	Student	Batch
1	Sancheeta	CSE
2	Navneet	CSE
3	Kartikay	ECE
4	Samyak	IT

Student Id	Course	Teacher	Grade
1	AI	Prof. Ajay	B
2	AI	Prof. Ajay	A
2	Database	Prof. Aman	B
3	OS	Prof. Deep	A
3	Database	Prof. Aman	B
4	OS	Prof. Deep	B

3NF

Id	Student	Batch
1	Sancheeta	CSE
2	Navneet	CSE
3	Kartikay	ECE
4	Samyak	IT

Course ID	Course	Teacher
1	AI	Prof. Ajay
2	Database	Prof. Aman
3	OS	Prof. Deep

Student Id	Course ID	Grade
1	1	B
2	1	A
2	2	B
3	3	A
3	2	B
4	3	B

BCNF

Id	Student	Batch
1	Sancheeta	CSE
2	Navneet	CSE
3	Kartikay	ECE
4	Samyak	IT

Course ID	Course	Teacher
1	AI	Prof. Ajay
2	Database	Prof. Aman
3	OS	Prof. Deep

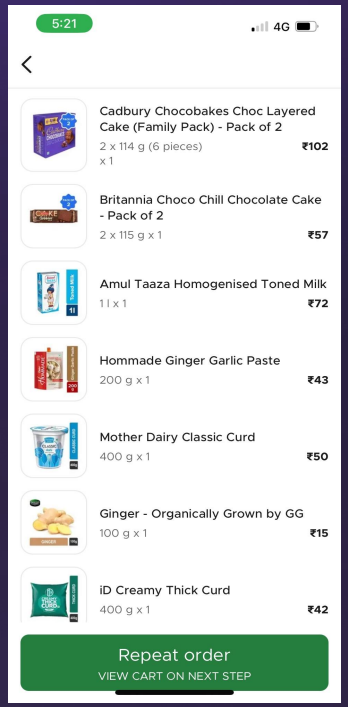
Student Id	Course ID	Grade
1	1	B
2	1	A
2	2	B
3	3	A
3	2	B
4	3	B

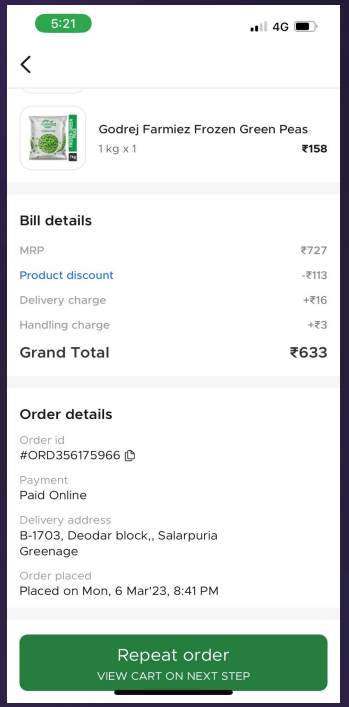
Benefits of Normalisation

- Reduces data redundancy which saves disk space
- Improves data consistency by putting data in one place
- Simplifies data maintenance for updation as data grows
- Enhances flexibility for changing requirements
- Improves query performance

Example

At Blinkit, we are planning to build an order system where the screen will look like the following screenshot. Let's build the tables using the normal forms we learnt - https://bit.ly/41q4brG





Airtribe

Thank You!

See you again on next date here

Airtribe