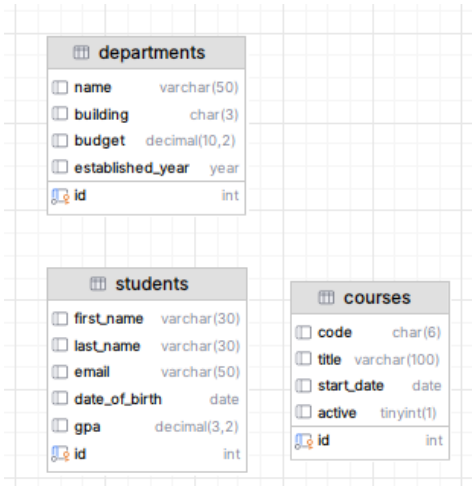


03. Data Types - SOLUTIONS

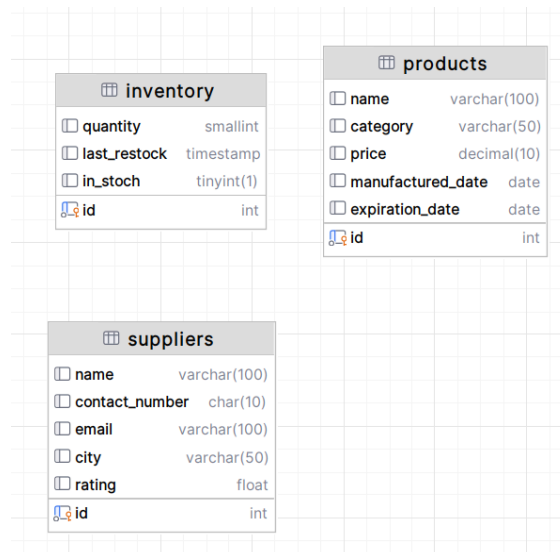
Task 1

1. Create a 3 database table to manage student information. Tables included should have data about students, data about departments and data about courses. You should not include relationships but you should include at least 5 different data types across all 3 tables. Data types need to be meaningful, meaning that you need to explain why exactly you decided to use specific data types.



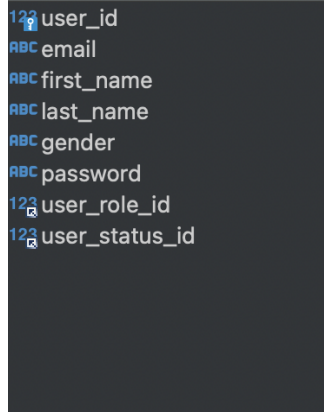
Task 2

2. Develop a database table for tracking product inventory. You need to come up with 3 different tables that need to be meaningful for the related topic, and you need to have at least 5 attributes in each table with as many different data types as possible.



Task 3

3. In the image below, a table of users is provided. You are provided with all attributes that are in the table. Recreate this table in MySQL and add all data types for every attribute. You will need to explain the reason behind your choice of data type.



```
123 user_id
ABC email
ABC first_name
ABC last_name
ABC gender
ABC password
123 user_role_id
123 user_status_id
```

Attribute	Data Type	Reason for Choice
user_id	INT UNSIGNED NOT NULL AUTO_INCREMENT	The INT type is suitable for storing numeric identifiers. Using UNSIGNED allows only positive numbers, which makes sense for IDs. The AUTO_INCREMENT option ensures each new user automatically receives a unique ID number.
email	VARCHAR(50)	Email addresses consist of characters and vary in length, so VARCHAR is the most appropriate choice. A length of 50 characters is enough for most standard email formats while keeping storage efficient.
first_name	VARCHAR(30)	Email addresses consist of characters and vary in length, so VARCHAR is the most appropriate choice. A length of 50 characters is enough for most standard email formats while keeping storage efficient.
last_name	VARCHAR(30)	Same reasoning as for first_name; VARCHAR(30) offers flexibility for different surname lengths.
gender	ENUM('Male', 'Female', 'Other')	The ENUM type is ideal when there is a limited and predefined set of possible values. It ensures that only these specific values can be stored, improving data consistency.
password	VARCHAR(255)	Passwords are stored as encrypted or hashed strings rather than plain text. Hash values can be quite long, so VARCHAR(255) ensures enough space for all possible encrypted formats while maintaining flexibility.
user_id_role	INT NOT NULL	Represents a reference to another table that defines user roles (e.g., admin, student, instructor). Using INT ensures efficient storage and quick joins when linking tables.
user_status_id	INT NOT NULL	Similar to user_role_id, this field links to another table that defines user statuses (e.g., active, inactive, suspended). Using INT is standard for identifiers and relationships.

Task 4

4. In the image below, a table of customers is provided. You are provided with all attributes that are in the table. Recreate this table in MySQL and add all data types for every attribute. You will need to explain the reason behind your choice of data type.

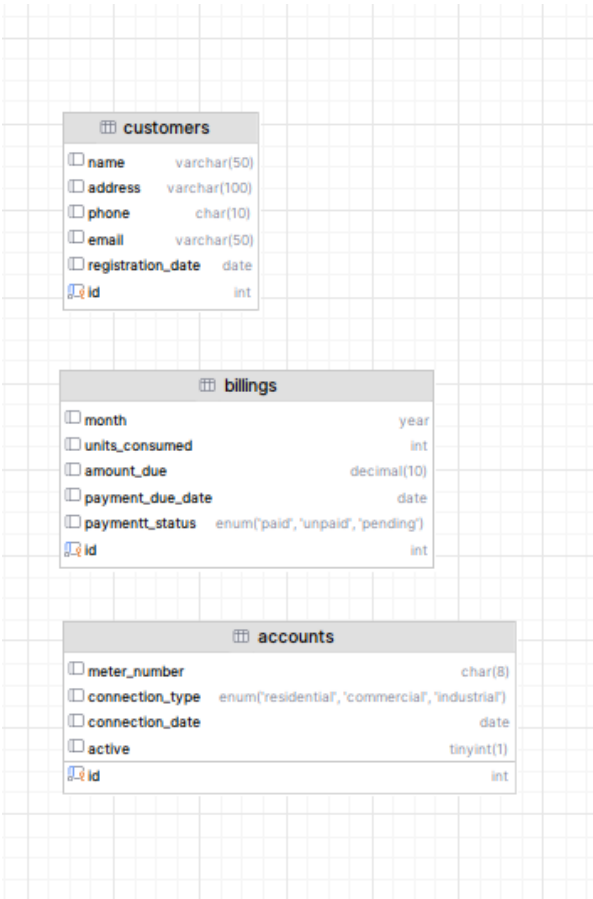
```
123 customerNumber
ABC customerName
ABC contactLastName
ABC contactFirstName
ABC phone
ABC addressLine1
ABC addressLine2
ABC city
ABC state
ABC postalCode
ABC country
123 salesRepEmployee
123 creditLimit
```

Attribute	Data Type	Reason for Choice
customerNumber	INT UNSIGNED NOT NULL AUTO_INCREMENT	Used as the unique identifier for each customer. INT efficiently stores numeric IDs, UNSIGNED ensures only positive values, and AUTO_INCREMENT automatically assigns new numbers to each record.
customerName	VARCHAR(50)	Length can vary, so VARCHAR is ideal. The limit of 50 characters accommodates most business names.
customerFirstName	VARCHAR(30)	Used to store the customer's first name. Names vary in length, and 30 characters is sufficient for most entries.
customerLasName	VARCHAR(30)	Stores the customer's last name. Similar reasoning as customerFirstName; 30 characters ensures enough space for longer surnames.
phone	VARCHAR(20)	Phone numbers may include symbols such as "+", "-", or spaces. Therefore, they are stored as text rather than numeric values. VARCHAR(20) is sufficient for international formats.
addressLine1	VARCHAR(100)	Stores the main part of the customer's street address. Addresses vary widely in length, so VARCHAR(100) provides flexibility.
addressLine2	VARCHAR(100)	Optional second line for apartment numbers, suite details, or additional address information. Using the same length as addressLine1 maintains consistency.
city	VARCAR(50)	City names are stored as text. A limit of 50 characters is more than enough for most locations.
status	VARCHAR(50)	Stores the name of the state, province, or region. VARCHAR(50) ensures flexibility across different countries.
postalCode	VARCHR(15)	Postal or ZIP codes can contain letters, numbers, and special characters (like "-" or spaces). Therefore, VARCHAR(15) is used instead of a numeric type.
country	VARCHAR(50)	Country names vary in length, so VARCHAR(50) is suitable for almost all cases.
salesRepEmployeeNumb er	INT NOT NULL	Represents the employee responsible for managing the customer. It typically references an employee table. Using INT ensures efficient linking between

		records.
creditLimit	DECIMAL(10,2)	Represents the customer's credit limit with up to 10 digits in total and 2 decimal places (for cents). The DECIMAL type ensures accurate financial calculations.

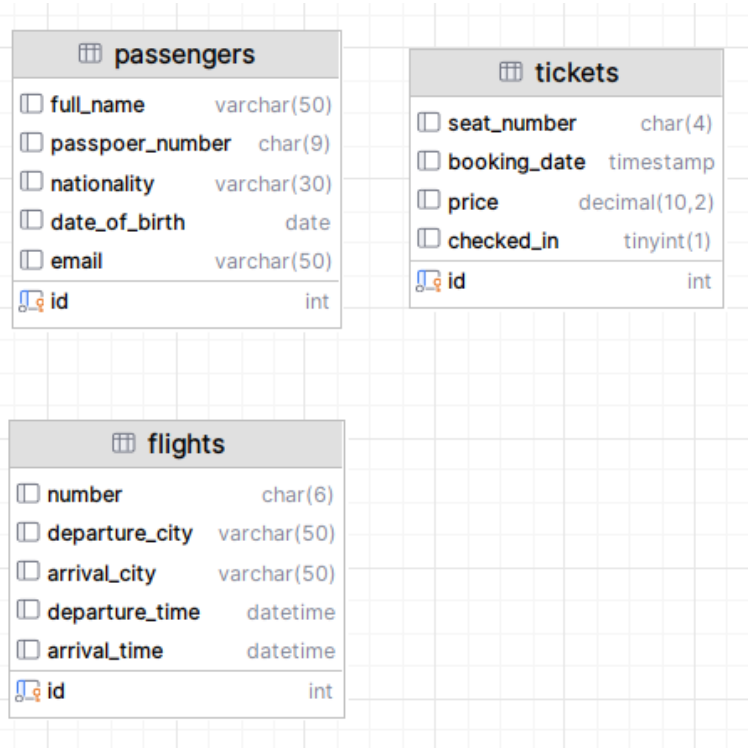
Task 5

5. Create three tables for an electricity bill management system. Tables included should have data about customers, account and billing. You should not include relationships but you should include at least 5 different data types across all 3 tables. Data types need to be meaningful, meaning that you need to explain why exactly you decided to use specific data types.



Task 6

- Develop a database for management of airplane tickets. You need to come up with 3 different tables that need to be meaningful for the related topic, and you need to have at least 5 attributes in each table with as many different data types as possible.



Task 7

- In the image below, a table of users is provided. You are provided with all attributes that are in the table. Recreate this table in MySQL and add all data types for every attribute. You will need to explain the reason behind your choice of data type.

123	address_id
ABC	address
ABC	address2
ABC	district
123	city_id
ABC	postal_code
ABC	phone
ABC	location
ABC	last_update

Attribute	Data Type	Reason for Choice
address_id	INT UNSIGNED NOT NULL AUTO_INCREMENT	Serves as the unique identifier for each address record. Using INT ensures efficient numeric storage, UNSIGNED restricts values to positive numbers, and AUTO_INCREMENT automatically generates new IDs for each record.
address	VARCHAR(100)	Stores the main street address (e.g., street name and number). The length of 100 characters provides flexibility for different address formats.
address2	VARCHAR(100)	Optional field for additional address details such as apartment number, suite, or building name. Having the same length as address keeps the table consistent.
district	VARCHAR(50)	Stores the district, region, or area name. The VARCHAR(50) type offers enough space for various administrative divisions.
city_id	INT NOT NULL	Refers to a related city record (foreign key). INT is efficient for indexing and joining with other tables.
postal_code	VARCHAR(15)	Postal or ZIP codes can include numbers, letters, and special characters (e.g., "-" or spaces). Therefore, VARCHAR(15) is more suitable than a numeric type.
phone	VARCHAR(20)	Phone numbers may include symbols such as "+", "()", and "-", so they are stored as text. The length of 20 supports both local and international formats.
location	POINT	Used to store geographic coordinates (latitude and longitude). The POINT type allows spatial queries, making it possible to locate or map addresses based on their exact geographic position.
last_update	DATETIME DEFAULT CURRENT_TIMESTAMP	Records the exact date and time when the record is last modified. The DATETIME type stores both date and time values, while DEFAULT CURRENT_TIMESTAMP automatically sets the current system time when a new record is inserted, ensuring accurate tracking.

Task 8

- In the image below, a table of employees is provided. You are provided with all attributes that are in the table. Recreate this table in MySQL and add all data types for every attribute. You will need to explain the reason behind your choice of data type

```

123 employeeNumber
ABC lastName
ABC firstName
ABC extension
ABC email
ABC officeCode
123 reportsTo
ABC jobTitle

```

Attribute	Data Type	Reason for Choice
employeeNumber	INT UNSIGNED NOT NULL AUTO_INCREMENT	Unique identifier for each employee. INT is efficient for storage, UNSIGNED ensures positive values, and AUTO_INCREMENT automatically generates IDs.
lastName	VARCHAR(50)	Stores the employee's last name. 50 characters is sufficient for most surnames.
firstName	VARCHAR(50)	Stores the employee's first name. Same length as lastName for consistency.
extension	VARCHAR(10)	Phone extension. VARCHAR allows numbers and special characters like "x" or "-".
email	VARCHAR(100)	Employee email. 100 characters accommodate most email formats.
officeCode	CHAR(10) NOT NULL	Foreign key referencing the office table. CHAR ensures fixed-length storage, good for codes.
reportsTo	INT NOT NULL	Foreign key referencing another employee (manager). INT is efficient for joins and indexing.
jobTitle	VARCHAR(50)	Employee job title. 50 characters are enough for most positions.