Overview

1. **Data Types for Attributes/Fields**
2. **DDL SQL Statements**
3. **Database Integrity**
4. **DML SQL Statements** (Insert, Update, Delete)

MySQL Data Types

TEXT Types (I & II)

- **CHAR(size)**: Fixed length string (max 255 chars).
- **VARCHAR(size)**: Variable length string (max 255 chars). Exceeding 255 converts it to `TEXT`.
- **TINYTEXT**: Max length 255 chars.
- **TEXT**: Max length 65,535 chars.
- **BLOB**: Binary Large Object (max 65,535 bytes).
- **MEDIUMTEXT**: Max length 16,777,215 chars.
- **MEDIUMBLOB**: Max 16,777,215 bytes.
- **LONGTEXT**: Max length 4,294,967,295 chars.
- **LONGBLOB**: Max 4,294,967,295 bytes.
- **ENUM(x, y, z, etc.)**: List of possible values (max 65,535).
- **SET**: Similar to ENUM but allows up to 64 list items.

Exact Number Types

- **TINYINT**: 1 byte, range: `-128 to 127` or `0 to 255 UNSIGNED`.
- **SMALLINT**: 2 bytes, range: `-32768 to 32767` or `0 to 65535 UNSIGNED`.
- **MEDIUMINT**: 3 bytes, range: `-8388608 to 8388607` or `0 to 16777215 UNSIGNED`.
- **INT**: 4 bytes.
- **BIGINT**: 8 bytes.
- **DECIMAL(size, d)**: Fixed-point, max 65 digits with precision after decimal.

Approximate Number Types

- **FLOAT(size, d)**: Small number with floating decimal (4 bytes).

- **DOUBLE(size, d) or REAL**: Large number with floating decimal (8 bytes).

Date & Time Types

- **DATE**: Format `YYYY-MM-DD`, range `1000-01-01` to `9999-12-31`.
- **TIME**: Format `HH:MM:SS`, range `-838:59:59` to `838:59:59`.
- **DATETIME**: Combines date and time (`YYYY-MM-DD HH:MM:SS`).
- **YEAR**: Two- or four-digit year.

Advice: Avoid using date data types for coursework; instead, define dates as integers (format `YYMMDD`).

Main DDL Statements

1. **CREATE TABLE**: Define a new table.

```
CREATE TABLE TableName (
  columnName dataType [NOT NULL] [UNIQUE] [DEFAULT defaultValue] [CHECK
condition]
  [PRIMARY KEY (columns)] [FOREIGN KEY (columns) REFERENCES
ParentTable(columns)]
  [ON UPDATE | ON DELETE referentialAction] [CHECK condition]
);
```

   - Example:

```
CREATE TABLE journey (
  ID INTEGER NOT NULL AUTO_INCREMENT PRIMARY KEY,
  DISTANCE INTEGER,
  JOURNEYDATE DATE,
  TICKET CHAR(1)
);
```

2. **ALTER TABLE**: Modify an existing table.
   - Add a column:

```
ALTER TABLE journey ADD time INTEGER;
```

   - Modify a column:

```
ALTER TABLE journey MODIFY COLUMN time TIME NOT NULL;
```

3. **DROP TABLE**: Remove a table from the database.

```
DROP TABLE dummy;
```

Database Integrity

Integrity constraints protect databases from becoming inconsistent. There are five main types:

1. **Required Data**: Ensure certain columns cannot take NULL values.
   - Use `NOT NULL` for mandatory attributes.
   - Example: `position VARCHAR(10) NOT NULL`.
2. **Domain Constraints**: Ensure data within a certain domain.
   - Example using `CHECK`:

     ```
     CHECK (gender IN ('M', 'F', 'X'))
     ```

   - Example using `ENUM`: `sex ENUM('M', 'F', 'X') NOT NULL`.
3. **Entity Integrity**: Primary keys must be unique and non-null.
   - Example: `PRIMARY KEY (branchNo)` or composite key `PRIMARY KEY (clientNo, propertyNo)`.
4. **Referential Integrity**: Foreign keys must refer to existing primary keys in another table.
   - Example:

     ```
     FOREIGN KEY (branchNo) REFERENCES Branch(branchNo)
     ```

5. **General Constraints**: Enforce complex rules.
   - Example:

     ```
     CREATE ASSERTION StaffNotHandlingTooMuch
     CHECK (NOT EXISTS (
       SELECT staffNo
       FROM PropertyForRent
       GROUP BY staffNo
     ```

```
  HAVING COUNT(*) > 100
));
```

DML SQL Statements

1. **INSERT**

```
INSERT INTO journey (distance, journeydate, ticket, time)
VALUES (2, '2017-03-14', 'o', '8:30:00');
```

2. **UPDATE**

```
UPDATE journey SET ticket='p' WHERE ID=1;
```

3. **DELETE**

```
DELETE FROM journey WHERE ID=1;
```

Key Takeaways

- **DDL (Data Definition Language)** is crucial for creating and modifying database structures.
- **DML (Data Manipulation Language)** is used for data operations like inserting, updating, and deleting records.
- **Database Integrity** is maintained through constraints like primary keys, foreign keys, and rules that prevent inconsistent data.