35.1 Relational model

Database models

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

A ***database model*** is a conceptual framework for database systems, with three parts:

- *Data structures* that prescribe how data is organized.
- *Operations* that manipulate data structures.
- *Rules* that govern valid data.

Some database models are described in academic literature and standardized by official organizations. Others are derived informally from prominent database systems.

The ***relational model*** is a database model based on a tabular data structure. The model was published in 1970 by E. F. Codd of IBM and released in commercial products around 1980. The data structure, operations, and rules are standardized in SQL, the universal query language of relational databases.

Many non-relational database models have been published and implemented in database systems. In the 1960s and 1970s, hierarchical and network databases were dominant. At the time, computers were relatively slow and memory was limited. As a result, these databases were optimized for performance at the expense of simplicity and flexibility. Relational databases are relatively simple to manage and, as performance improved during the 1980s, rapidly displaced hierarchical and network databases.

Relational databases were initially designed for transactional data, such as bank transactions and airline reservations. The rise of the internet in the 1990s generated ***big data***, characterized by unprecedented data volumes and rapidly changing data structures. Many alternative database models and systems, optimized for big data, have appeared since 2000. However, relational databases have gradually improved support for big data and continue to dominate the commercial database market.

Table 35.1.1: Example database models.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

	Primary data structure	Initial product releases	Example database system	Strengths
Hierarchical	Tree	1960s	IMS	Fast queries Efficient storage
Network	Linked list	1970s	IDMS	Fast queries Efficient storage

Relational	Table	1980s	Oracle Database	Productivity and simplicity Transactional applications
Object	Class	1990s	ObjectStore	Integration with object-oriented programming languages
Graph	Vertex and edge	2000s	Neo4j	Flexible schema Evolving business requirements
Document	XML JSON	2010s	MongoDB	Flexible schema Unstructured and semi-structured data

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

PARTICIPATION ACTIVITY 35.1.1: Database models.

1) Which database is relational?
   - Oracle Database
   - IDMS
   - MongoDB

2) The relational model was originally developed for which types of applications?
   - Big data storage and analysis
   - Transactional applications like banking and airline reservations
   - Desktop applications with small databases

3) What was the initial impediment to commercial adoption of relational databases in the early 1980s?
   - Reliability
   - Processing speed
   - Cost

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Relational data structure

The relational data structure is based on set theory. A ***set*** is an unordered collection of elements enclosed in braces. Ex: *{a, b, c}* and *{c, b, a}* are the same, since sets are not ordered. A ***tuple*** is an ordered collection of elements enclosed in parentheses. Ex: *(a, b, c)* and *(c, b, a)* are different, since tuples are ordered.

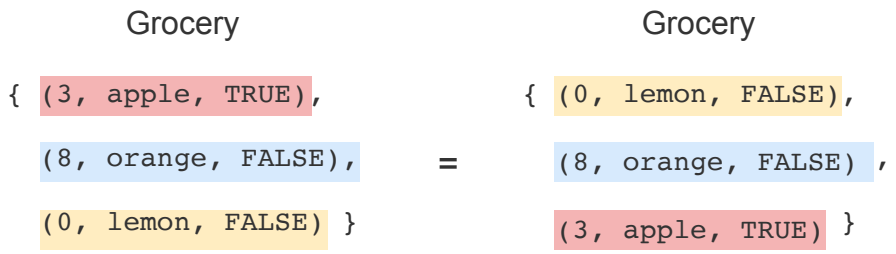
The data structure organizes data in tables:

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

- A ***table*** has a name, a fixed tuple of columns, and a varying set of rows.
- A ***column*** has a name and a data type.
- A ***row*** is an unnamed tuple of values. Each value corresponds to a column and belongs to the column's data type.
- A ***data type*** is a named set of values, from which column values are drawn.

Since a table is a set of rows, the rows have no inherent order.

PARTICIPATION ACTIVITY 35.1.2: Table rows are not ordered.



Animation content:

Step 1: The Grocery table is set of three rows. A table named Grocery appears with three rows:
3, apple, TRUE
8, orange, FALSE
0, lemon, FALSE

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Step 2: Since sets are not ordered , the left and right tables are the same. A second table also named Grocery appears to the right of the first table, with rows in a different order:
0, lemon, FALSE
8, orange, FALSE
3, apple, TRUE
The first row of the first Grocery table and the third row of the second Grocery table are highlighted

red. The second row of the first Grocery table and the second row of the second Grocery table are highlighted blue. The third row of the first Grocery table and the first row of the second Grocery table are highlighted yellow. An equals sign appears between the two tables.

Animation captions:

1. The Grocery table is set of three rows.
2. Since sets are not ordered , the left and right tables are the same.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

The terms table, column, row, and data type are commonly used in database processing. Relation, attribute, tuple, and domain are equivalent mathematical terms, often used in academic literature. File, field, record, and data type are similar terms from file processing.

Table 35.1.2: Similar data structure terms.

Databases	Mathematics	Files
Table	Relation	File
Column	Attribute	Field
Row	Tuple	Record
Data type	Domain	Data type

PARTICIPATION ACTIVITY 35.1.3: Relational data structure.

1) Which terms are commonly used in database processing?

- Tuple, relation, attribute
- Row, table, column
- Record, file, field

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

2) Are these tables the same?

{ (8, mango, FALSE), (-11, watermelon, FALSE) }
{ (-11, watermelon, FALSE), (8, mango, FALSE) }

- Yes
- No

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

3) In the relational data structure, which components are named?

- Data type, row, table
- Data type, table
- Data type, table, column

4) Can a query select one specific row from a table?

- Yes, by specifying the row name
- Yes, by specifying one or more row values
- No

Relational operations

Like the relational data structure, relational operations are based on set theory. Each operation generates a result table from one or two input tables:

- *Select* selects a subset of rows of a table.
- *Project* eliminates one or more columns of a table.
- *Product* lists all combinations of rows of two tables.
- *Join* combines two tables by comparing related columns.
- *Union* selects all rows of two tables.
- *Intersect* selects rows common to two tables.
- *Difference* selects rows that appear in one table but not another.
- *Rename* changes a table name.
- *Aggregate* computes functions over multiple table rows, such as sum and count.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

These operations are collectively called ***relational algebra*** and are the theoretical foundation of the SQL language. Since the result of relational operations is always a table, the result of an SQL query is also a table.

PARTICIPATION ACTIVITY 35.1.4: Relational operations and SQL.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Select

```
SELECT *
FROM Employee
WHERE Salary > 50000;
```

Product

```
SELECT *
FROM Employee, Department;
```

Project

```
SELECT Name
FROM Employee;
```

Join

```
SELECT *
FROM Employee, Department
WHERE Employee.DeptCode =
      Department.DeptCode;
```

Animation content:

Step 1: SELECT * selects all columns in the Employee table. SELECT * FROM Employee WHERE Salary > 50000; appears. SELECT * is highlighted.

Step 2: The select operation selects only rows for which the Salary column is > 50000. The caption Select appears. WHERE Salary > 50000 is highlighted.

Step 3: The project operation selects only the Name column. The caption Project appears. SELECT Name FROM Employee; appears. SELECT Name is highlighted.

Step 4: The product operation selects all combinations of Employee and Department rows. The caption Product appears. SELECT * FROM Employee, Department; appears. FROM Employee, Department is highlighted.

Step 5: The join operation combines Employee and Department by comparing the tables' DepartCode columns. The caption Join appears. SELECT * FROM Employee, Department WHERE Employee.DepartmenCode = Department.DepartCode; appears. WHERE Employee.DepartmenCode = Department.DepartCode is highlighted.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Animation captions:

1. SELECT * selects all columns in the Employee table.

2. The select operation selects only rows for which the Salary column is > 50000.
3. The project operation selects only the Name column.
4. The product operation selects all combinations of Employee and Department rows.
5. The join operation combines Employee and Department by comparing the tables' DepartCode columns.

PARTICIPATION ACTIVITY 35.1.5: Relational operations.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

1) What is the result of a relational operation?
   - A row
   - A column
   - A table

2) Name three relational operations.
   - Select, project, and union
   - Square root, exponent, and logarithm
   - Integrate and differentiate

3) An SQL statement can implement only one relational operation.
   - True
   - False

Relational rules

Rules are logical constraints that ensure data is valid.

Relational rules are part of the relational model and govern data in every relational database. Ex:

- *Unique primary key*. All tables have a primary key column, or group of columns, in which values may not repeat.
- *Unique column names*. Different columns of the same table have different names.
- *No duplicate rows*. No two rows of the same table have identical values in all columns.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Business rules are based on business policy and specific to a particular database. Ex: All rows of the Employee table must have a valid entry in the DepartCode column. Ex: PassportNumber values may not repeat in different Employee rows.

Relational rules are implemented as SQL ***constraints*** and enforced by the database system. Business rules are discovered during database design and, like relational rules, often implemented as SQL constraints. However, some complex business rules must be enforced by applications running on the database.

PARTICIPATION ACTIVITY 35.1.6: Business rule example.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Employee

ID	Name	Salary
2538	Lisa Ellison	45000
~~5384~~	~~Sam Snead~~	~~30500~~
6381	Maria Rodriguez	92300

Task

EmployeeID	TaskName
2538	Fix software bug
~~5384~~	~~Write annual report~~
~~5384~~	~~Submit timesheet~~

```
CREATE Table Task (
   ...
   FOREIGN KEY (EmployeeID) REFERENCES Employee (ID)
      ON DELETE CASCADE
   ...
);
```

Animation content:

Step 1: Sam Snead has two tasks. Two tables appear named Employee and Task. Employee has three columns named ID, Name, and Salary. Task has two columns named EmployeeID and TaskName. Row two of Employee is highlighted with these values:
5384, Sam Snead, and 30500
Rows two and three Task are highlighted with these values:
5384, Write annual report
5384, Submit timesheet

Step 2: A business rule requires that, when an employee is deleted, the employee's tasks are also deleted. A red line strikes through the highlighted rows.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Step 3: The business rule is implemented as an SQL constraint. A CREATE TABLE statement for Task appears. Within the statement, the following clause is highlighted: FOREIGN KEY (EmployeeID) REFERENCES Employee (ID) ON DELETE CASCADE.

Animation captions:

1. Sam Snead has two tasks.

2. A business rule requires that, when an employee is deleted, the employee's tasks are also deleted.
3. The business rule is implemented as an SQL constraint.

PARTICIPATION ACTIVITY 35.1.7: Relational rules.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

1) Unique primary key is an example of a relational rule.
   - True
   - False

2) Delete cascade is an example of a relational rule.
   - True
   - False

3) Data in a relational database can violate relational rules.
   - True
   - False

CHALLENGE ACTIVITY 35.1.1: Relational model.

539740.3879454.qx3zqy7

Start

Different terms are used for similar concepts in databases, mathematics, and file systems. Select the correct database term corresponding to the following terms.

Mathematics	Databases
Tuple	Pick
Relation	Pick

File systems	Databases
Record	Pick
Data type	Pick

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

1 2

Check Next

Exploring further:

- Database models (Wikipedia)
- Original paper on the relational model, by E. F. Codd

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

35.2 Structured Query Language

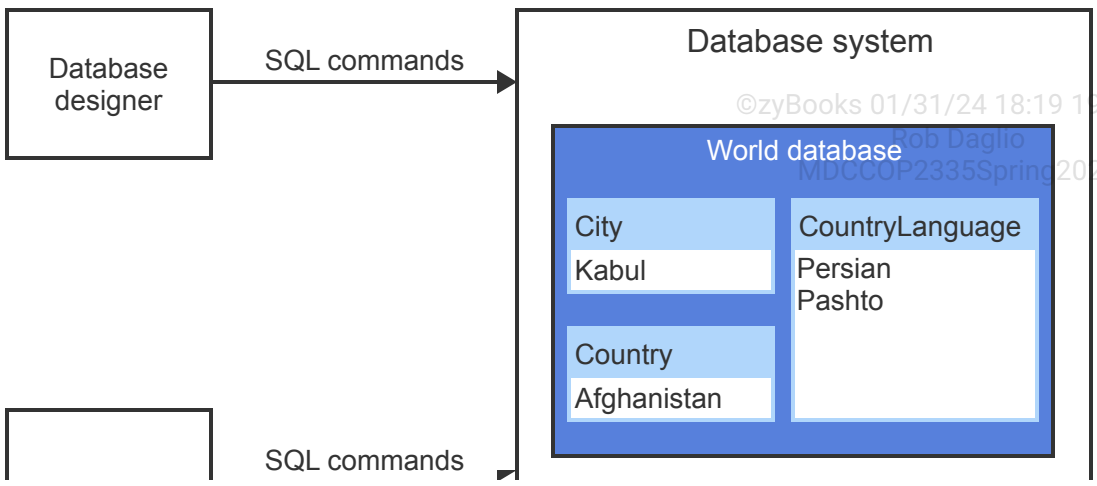
Structured Query Language

Structured Query Language (***SQL***) is a high-level computer language for storing, manipulating, and retrieving data. SQL is the standard language for relational databases, and is commonly supported in non-relational databases. SQL is pronounced either 'S-Q-L' or 'seekwəl', but the preferred pronunciation is 'S-Q-L'.

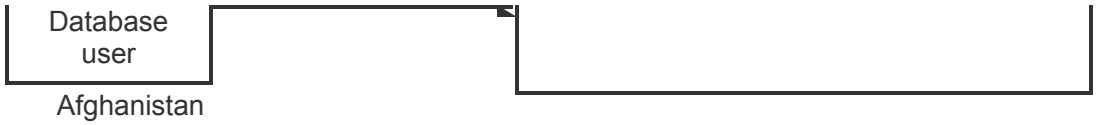
The SQL standard is published jointly by the American National Standards Institute (ANSI) and the International Organization for Standardization (ISO). The standard was first published in 1986 and has since evolved through many versions. For details on the standard, see the link in Exploring Further, below.

Relational databases generally support most important elements of the SQL standard. However, most relational databases do not support the entire standard and many support custom extensions. This material describes standard SQL in most cases. Where MySQL differs from the standard, this material describes MySQL syntax and calls out the differences.

PARTICIPATION ACTIVITY 35.2.1: SQL interaction with a database system.



©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024



Animation content:

Step 1: A database designer uses SQL to create a database and the database tables. A small box named Database designer appears on the left and a large box named Database system appears on the right. An arrow named SQL commands appears going from the Database designer box to the Database system box. The words CREATE DATABASE appear next to the Database designer box and move along the SQL commands arrow to the Database system box. A new box named World database appears inside the Database system box. The words CREATE TABLE appear next to the Database designer box and move along the SQL commands arrow to the Database system box. A new table named City appears inside the World database box. The words CREATE TABLE appear next to the Database designer box and move along the SQL commands arrow to the Database system box. A new table named Country appears inside the World database box. The words CREATE TABLE appear next to the Database designer box and move along the SQL commands arrow to the Database system box. A new table named CountryLanguage appears inside the World database box.

Step 2: A database user uses SQL to insert, retrieve, update, and delete data from the tables. A small box named Database user appears to the left of the Database system box. An arrow named SQL commands appears from the Database user box to the Database system box. The words INSERT Kabul appear next to the Database user box and move along the SQL commands arrow to the Database system box. Kabul appears in row one of the City table. The words SELECT Afghanistan appear next to the Database user box and move along the SQL commands arrow to the Database system box. Afghanistan then appears in row one of the Country table and then moves underneath the Database user box.

Animation captions:

1. A database designer uses SQL to create a database and the database tables.
2. A database user uses SQL to insert, retrieve, update, and delete data from the tables.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

PARTICIPATION ACTIVITY 35.2.2: SQL.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

1) SQL commands can create databases and tables.

   - True
   - False

2) A database designer and database user both use SQL.

- True
- False

3) All database systems use identical SQL statements.

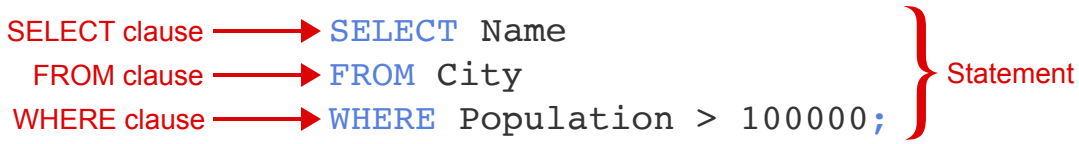
- True
- False

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

SQL syntax

An SQL ***statement*** is a complete command composed of one or more clauses. A ***clause*** groups SQL keywords like SELECT, FROM, and WHERE with table names like City, column names like Name, and conditions like Population > 100000. *An SQL statement may be written on a single line, but good practice is to write each clause on a separate line.*

PARTICIPATION ACTIVITY 35.2.3: Three clauses in a SELECT statement.



Animation content:

Step 1: The SELECT clause starts the statement. Name is a column name. The code SELECT Name appears. Text appears to the left of the code and states SELECT clause. An arrow appears from this text to the code.

Step 2: The FROM clause must follow the SELECT clause. City is a table name. The code FROM City appears below the previous code. Text appears below the previous text and states FROM clause. An arrow appears from this text to the second line of code.

Step 3: The WHERE clause is optional. When included, the WHERE clause must follow the FROM clause. Population > 100000 is a condition. The code WHERE Population is greater than 100000 appears below the previous code. Text appears below the previous text and states WHERE clause. An arrow appears from this text to the third line of code.

Step 4: The three clauses ending in a semicolon comprise a statement. A semicolon is added to the end of the third line of code and a right bracket encompassing all three lines of code appears with the text Statement appearing to the right of the bracket.

Animation captions:

1. The SELECT clause starts the statement. Name is a column name.
2. The FROM clause must follow the SELECT clause. City is a table name.
3. The WHERE clause is optional. When included, the WHERE clause must follow the FROM clause. Population > 100000 is a condition.
4. The three clauses ending in a semicolon comprise a statement.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

In MySQL, all SQL statements end with a semicolon. SQL keywords like SELECT, FROM, WHERE, etc. are not case sensitive. Ex: SELECT and select are equivalent. However, identifiers like column names and table names are case sensitive in many database systems. This material uses capital letters for SQL keywords so the keywords stand out from other syntactic parts. The table below summarizes various syntactic features of SQL.

Table 35.2.1: Summary of SQL syntax features.

Type	Description	Examples

Literals	Explicit values that are string, numeric, or binary.
	Strings must be surrounded by single quotes or double quotes.
	Binary values are represented with `x'0'` where the 0 is any hex value.
		`'String'`
		`"String"`
		`123`
		`x'0fa2'`

Keywords	Words with special meaning.	SELECT, FROM, WHERE
Identifiers	Objects from the database like tables, columns, etc.	City, Name, Population
Comments	Statement intended only for humans and ignored by the database when parsing an SQL statement.	`-- single line comment` `/* multi-line Comment */`

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

All database systems recognize single quotes for string literals, but some also recognize double quotes. This material uses single quotes to ensure the SQL statements are compatible with all database systems.

PARTICIPATION ACTIVITY 35.2.4: SQL syntax.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

1) The INSERT statement adds a student to the Student table. How many clauses are in the INSERT statement?

```
INSERT INTO Student
VALUES (888, 'Smith', 'Jim',
3.0);
```

- 1
- 2
- 3

2) The SQL statement below is used to select students with the last name "Smith". What is wrong with the statement?

```
SELECT FirstName
FROM Student
WHERE LastName = Smith;
```

- The literal "Smith" must be surrounded by single quotes or double quotes.
- The WHERE clause should be removed.
- The last name "Smith" may not exist in the database.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

3) What is wrong with the SQL statement below?

```
SELECT FirstName
from Student
```

- The keyword from must be written FROM.
- The WHERE clause is missing.
- A terminating semicolon is missing.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

4) What is wrong with the SQL statement below?

```
SELECT Gpa
--FROM Student;
```

- The FROM clause is a comment.
- The WHERE clause is missing.
- Gpa should be a table name.

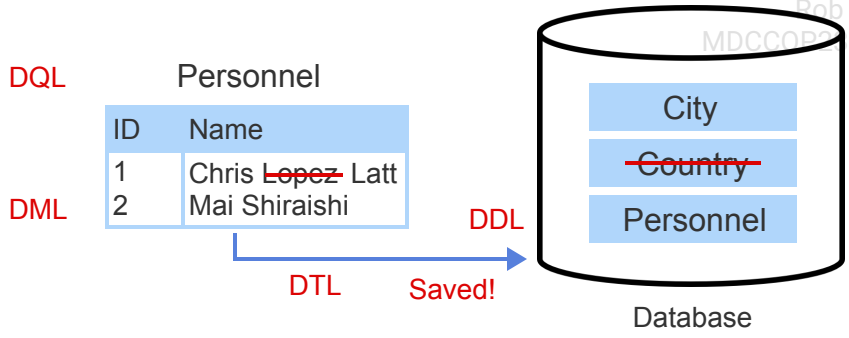
SQL sublanguages

The SQL language is divided into five sublanguages:

- ***Data Definition Language*** (DDL) defines the structure of the database.
- ***Data Query Language*** (DQL) retrieves data from the database.
- ***Data Manipulation Language*** (DML) manipulates data stored in a database.
- ***Data Control Language*** (DCL) controls database user access.
- ***Data Transaction Language*** (DTL) manages database transactions.

PARTICIPATION ACTIVITY 35.2.5: SQL sublanguages.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024



DCL

Username	Role
User 1	Administrator
~~User 2~~	~~User~~
User 3	User

Animation content:

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Step 1: DDL creates, alters, and drops tables. A cylinder named Database appears containing the words City and Country. The caption DDL appears. A line strikes through Country and a new word Personnel appears below Country.

Step 2: DQL selects data from a table. The caption DQL appears. The Personnel table appears next to the caption with one row.

Step 3: DML inserts, updates, and deletes data in a table. The caption DML appears. The values of the row in Personnel change and a new row is inserted.

Step 4: DCL grants and revokes permissions to and from users. The caption DCL appears next to a table with columns Username and Role. The table has three rows. A line strikes through the second row.

Step 5: DTL commits data to a database, rolls back data from a database, and creates savepoints. The caption DTL appears. An arrow appears from the Personnel table to the Database cylinder. The arrow is labeled saved.

Animation captions:

1. DDL creates, alters, and drops tables.
2. DQL selects data from a table.
3. DML inserts, updates, and deletes data in a table.
4. DCL grants and revokes permissions to and from users.
5. DTL commits data to a database, rolls back data from a database, and creates savepoints.

PARTICIPATION ACTIVITY 35.2.6: SQL sublanguages.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Match the SQL sublanguage to the statement behavior.

If unable to drag and drop, refresh the page.

DTL DDL DML DCL DQL

	Insert a data row into table product.
	Rollback database changes.
	Select all rows from table Product.
	Grant all permissions to user 'tester'.
	Create table Product.

Reset

Rob Daglio 1/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Exploring further:

- SQL standard (Wikipedia)

35.3 Managing databases

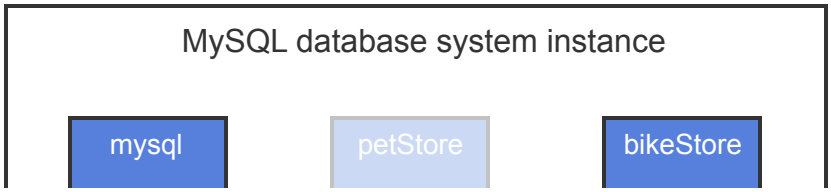
CREATE DATABASE and DROP DATABASE statements

A ***database system instance*** is a single executing copy of a database system. Personal computers usually run just one instance of a database system. Shared computers, such as computers used for cloud services, usually run multiple instances of a database system. Each instance usually contains multiple system and user databases.

Several SQL statements help database administrators, designers, and users manage the databases on an instance. ***CREATE DATABASE DatabaseName*** creates a new database. ***DROP DATABASE DatabaseName*** deletes a database, including all tables in the database.

PARTICIPATION ACTIVITY 35.3.1: CREATE DATABASE and DROP DATABASE statements.

Rob Daglio 1/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024



```
CREATE DATABASE petStore;
CREATE DATABASE bikeStore;
DROP DATABASE petStore;
```

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Animation content:

Static figure:
A rectangle, named MySQL database system instance, contains three smaller rectangles. The smaller rectangles are named mysql, petStore, and bikeStore. Three SQL statements appear:
Begin SQL code:
CREATE DATABASE petStore
CREATE DATABASE bikeStore
DROP DATABASE petStore
End SQL code.

Step 1: The mysql database is automatically installed by the MySQL database system and used for system administration. The mysql rectangle appears inside MySQL database system instance.

Step 2: The CREATE DATABASE statement creates a new database called petStore. The first SQL statement appears. The petStore rectangle appears inside MySQL database system instance.

Step 3: Additional databases can be created. The second SQL statement appears. The bikestore rectangle appears inside MySQL database system instance.

Step 4: The DROP DATABASE statement deletes the petStore database. The third SQL statement appears. The petStore rectangle fades out.

Animation captions:

1. The mysql database is automatically installed by the MySQL database system and used for system administration.
2. The CREATE DATABASE statement creates a new database called petStore.
3. Additional databases can be created.
4. The DROP DATABASE statement deletes the petStore database.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

PARTICIPATION ACTIVITY 35.3.2: CREATE DATABASE and DROP DATABASE statements.

Assume the database system currently contains only one user database, called 'university', in addition to system databases such as 'mySQL'.

1) The statement `CREATE DATABASE auto;` creates a database called 'auto'.
   - True
   - False

2) The statement `CREATE DATABASE university;` creates a second university database.
   - True
   - False

3) The statement `DROP DATABASE university;` deletes the university database and all associated tables.
   - True
   - False

4) The statement `DROP DATABASE nonprofit;` deletes the nonprofit database.
   - True
   - False

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

USE and SHOW statements

USE DatabaseName selects a default database for use in subsequent SQL statements.

Several SHOW statements provide information about databases, tables, and columns:

- ***SHOW DATABASES*** lists all databases in the database system instance.
- ***SHOW TABLES*** lists all tables in the default database.
- ***SHOW COLUMNS FROM TableName*** lists all columns in the TableName table of the default database.
- ***SHOW CREATE TABLE TableName*** shows the CREATE TABLE statement for the TableName table of the default database.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Additional SHOW statements generate information about system errors, configuration, privileges, logs, and so on. For details, see the link in Exploring Further, below.

PARTICIPATION ACTIVITY 35.3.3: USE and SHOW statements.

```
SHOW DATABASES;
```

Database
bikeStore
petStore
world

```
USE world;

SHOW TABLES;
```

Tables_in_world
City
Country
CountryLanguage

```
SHOW COLUMNS
FROM CountryLanguage;
```

Field	Type	Null	Key	Default	Extra
CountryCode	char(3)	NO	PRI		
Language	char(30)	NO	PRI		
IsOfficial	enum('T', 'F')	NO		F	
Percentage	float(4,1)	NO		0.0	

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Animation content:

Step 1: SHOW DATABASES lists 3 databases in a database system: bikeStore, petStore, and world. The code SHOW DATABASES; appears. A table appears with one column named Database and three rows:
bikeStore
petStore
world

Step 2: USE world selects the database called world. The code USE world; appears. Row three of the table is highlighted, containing the value world.

Step 3: SHOW TABLES lists the 3 tables in database world: City, Country, and CountryLanguage. The code SHOW TABLES; appears. A new table appears with one column named Tables_in_world and three rows:
City
Country
CountryLanguage

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Step 4: SHOW COLUMNS lists the 4 columns in table CountryLanguage. The column name, data type, and other column characteristics are listed. Two new lines of code appear:
Begin SQL code:
SHOW COLUMNS
FROM CountryLanguage;

End SQL code.
Row three of column Tables_in_world is highlighted, containing the value CountryLanguage. A new table appears with six columns named Field, Type, Null, Key, Default, and Extra. The new table has one row for each column in the CountryLanguage table.

Animation captions:

1. SHOW DATABASES lists 3 databases in a database system: bikeStore, petStore, and world.
2. USE world selects the database called world.
3. SHOW TABLES lists the 3 tables in database world: City, Country, and CountryLanguage.
4. SHOW COLUMNS lists the 4 columns in table CountryLanguage. The column name, data type, and other column characteristics are listed.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

The world database is from MySQL.com

PARTICIPATION ACTIVITY 35.3.4: USE and SHOW statements.

Refer to the animation above.

1) Which statement shows all databases in a database system?

- SHOW TABLES;
- SHOW DATABASES;
- SHOW COLUMNS;

2) Which statement shows all tables in the database petStore?

- SHOW TABLES;
- SHOW TABLES FROM petStore;
- SHOW DATABASES;

3) Which statement must precede a SHOW TABLES statement to see the tables from the bikeStore database?

- SHOW DATABASES;
- USE bikeStore;
- SHOW COLUMNS FROM bikeStore;

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

4) Which statement shows all the columns in the Country table?

- SHOW COLUMNS;
- SHOW COLUMNS Country;
- SHOW COLUMNS FROM Country;

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Exploring further:

- Creating and using MySQL databases
- MySQL SHOW statement

35.4 Tables

Tables, columns, and rows

All data in a relational database is structured in tables:

- A ***table*** has a name, a fixed sequence of columns, and a varying set of rows.
- A ***column*** has a name and a data type.
- A ***row*** is an unnamed sequence of values. Each value corresponds to a column and belongs to the column's data type.
- A ***cell*** is a single column of a single row.

A table must have at least one column but any number of rows. A table without rows is called an ***empty table***.

PARTICIPATION ACTIVITY 35.4.1: Tables, columns, and rows.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Employee

Column 1	Column 2	Column 3	
ID	Name	Salary	Column names
2538	Lisa Ellison	45000	Row 1

5384	Sam Snead	30500	Row 2
6381	Maria Rodriguez	92300	Row 3

Animation content:

Step 1: The Employee table has 3 columns with the names ID, Name, and Salary. The table has 3 rows. The Employee table appears with values:
2538, Lisa Ellison, 45000
5384, Sam Snead, 30500
6381, Maria Rodriguez, 92300

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Step 2: Each value appears in a single cell. In the first row of the table, "2538" is the ID column's value. The first row of column ID is highlighted.

Step 3: In the first row of the table, "Lisa Ellison" is the Name column's value. The first row of column Name is highlighted.

Step 4: In the first row of the table, "45000" is the Salary column's value. The first row of column Salary is highlighted.

Animation captions:

1. The Employee table has 3 columns with the names ID, Name, and Salary. The table has 3 rows.
2. Each value appears in a single cell. In the first row of the table, "2538" is the ID column's value.
3. In the first row of the table, "Lisa Ellison" is the Name column's value.
4. In the first row of the table, "45000" is the Salary column's value.

PARTICIPATION ACTIVITY 35.4.2: Tables, columns, and rows.

Refer to the Employee table in the above animation.

1) Which choice is a column name?

- 6381
- Sam Snead
- Salary

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

2) Which choice is a cell value?

- ID
- Elsa Brinks
- 30500

3) What are the components of a column?

- Name only
- Data type only
- Name and data type

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

4) Must a table have at least one row?

- Yes
- No

5) How does a database administrator specify column names and data types?

- In a special file managed by the database administrator
- With the Python language
- With the SQL language

Rules governing tables

Tables must obey relational rules, including:

1. *Exactly one value per cell*. A cell may not contain multiple values. Unknown data is represented with a special NULL value.
2. *No duplicate column names*. Duplicate column names are allowed in different tables, but not in the same table.
3. *No duplicate rows*. No two rows may have identical values in all columns.
4. *No row order*. Rows are not ordered. The organization of rows on a storage device, such as a disk drive, never affects query results.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Rules 3 and 4 follow directly from the definition of a table. A table is a set of rows. Since a set's elements may not repeat and are not ordered, the same is true of a table's rows.

Rule 4 is called ***data independence***. Data independence allows database administrators to improve query performance by changing the organization of data on storage devices, without affecting query results.

PARTICIPATION ACTIVITY 35.4.3: Rules governing tables.

Employee

ID	Name	Salary
2538	Lisa Ellison	45000
5384	Sam Snead	30500
6381	Maria Rodriguez	92300, ~~70000~~

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Employee

ID	Name	Salary	~~Salary~~
2538	Lisa Ellison	45000	~~35000~~
5384	Sam Snead	30500	~~25600~~
6381	Maria Rodriguez	92300	~~88100~~

Employee

ID	Name	Salary
2538	Lisa Ellison	45000
5384	Sam Snead	30500
~~5384~~	~~Sam Snead~~	~~30500~~

Animation content:

Step 1. A table cannot have multiple values in the same cell. The Employee table appears, with three columns named ID, Name, and Salary. Two values appear within the cell for the Salary column of row three:
92300, 70000
The second value, 70000, is struck out with a red line.

Step 2. A table cannot have multiple columns with the same name. The Employee table appears, with four columns named ID, Name, Salary, and Salary. The column name and all values in the second Salary column are struck out with red lines.

Step 3. A table cannot have multiple rows with identical data. The Employee table appears, with three columns named ID, Name, and Salary. The table has three rows. Rows two and three have identical values. Row three is struck out with a red line.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Animation captions:

1. A table cannot have multiple values in the same cell.
2. A table cannot have multiple columns with the same name.
3. A table cannot have multiple rows with identical data.

Duplicate rows

Most databases allow tables with duplicate rows, in violation of rule 3. However, these tables are usually temporary. Ex: External data with duplicate rows might be loaded to a temporary table. The duplicate rows are normally eliminated as data is moved to a permanent table.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

PARTICIPATION ACTIVITY 35.4.4: Rules governing tables.

1) _____ may appear in each cell.
   - Any number of values
   - Exactly one value
   - No values or one value

2) Where are duplicate column names allowed?
   - Within a single table.
   - In different tables.
   - Never.

3) What does the principle of data independence state?
   - The performance of a query is not related to the physical organization of data.
   - Data in each row is independent of data in all other rows.
   - The result of a database query is not affected by the physical organization of data on storage devices.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

4) How can driver's license information be added to the Employee table?

- Driver's licenses cannot be added, since the Employee table already has an ID column.
- By creating another column named ID.
- By creating another column with a new name, such as ID2 or DriverLicense.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

CREATE TABLE and DROP TABLE statements

The ***CREATE TABLE*** statement creates a new table by specifying the table name, column names, and column data types. Example data types are:

- *INT or INTEGER* — integer values
- *VARCHAR(N)* — values with 0 to N characters
- *DATE* — date values
- *DECIMAL(M, D)* — numeric values with M digits, of which D digits follow the decimal point

The ***DROP TABLE*** statement deletes a table, along with all the table's rows, from a database.

Figure 35.4.1: CREATE TABLE and DROP TABLE statements.

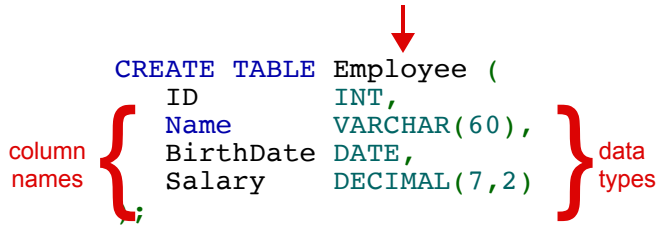
```
CREATE TABLE TableName
(
  Column1 DATA_TYPE,
  Column2 DATA_TYPE,
  ...
  ColumnN DATA_TYPE
);


DROP TABLE TableName;
```

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

PARTICIPATION ACTIVITY 35.4.5: Creating an Employee table.

table name



©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Animation content:

Static figure:
Begin SQL code:

```
CREATE TABLE Employee (
  ID INT,
  Name VARCHAR(60),
  BirthDate DATE,
  Salary DECIMAL(7,2)
);
```

End SQL code.

Step 1: The CREATE TABLE statement names the new table Employee. The first and last lines of code appear. The caption "table name" appears with an arrow pointing to Employee in the first line of code. An empty table named Employee appears to the right of the code.

Step 2: The column names and data types are separated by commas. Code lines two through five appear. The words ID, Name, BirthDate, and Salary are labeled "column names". The words SMALLINT, VARCHAR(60), DATE, and DECIMAL(7, 2) are labeled "data types". The table Employee gets four columns named ID, Name, BirthDate, and Salary.

Animation captions:

1. The CREATE TABLE statement names the new table "Employee".
2. The column names and data types are separated by commas.

PARTICIPATION ACTIVITY 35.4.6: Creating and deleting tables.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Refer to the animation above.

1) The Employee table is created with 4 columns.

- True
- False

2) The Name column contains character string values.

- True
- False

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

3) The statement `DROP Employee;` deletes the Employee table.

- True
- False

4) The DROP TABLE statement does not delete the table unless the table is empty.

- True
- False

PARTICIPATION ACTIVITY 35.4.7: Create a Product table with common data types.

Write the SQL to create a Product table with the following columns:

- *ID* - Integer
- *Name* - Variable-length string with maximum 40 characters
- *ProductType* - Variable-length string with maximum 3 characters
- *OriginDate* - Year, month, and day
- *Weight* - Decimal number with six significant digits and one digit after the decimal point

Place your CREATE TABLE statement before the INSERT and SELECT statements. Run your solution and verify the result table contains the three inserted rows.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Hint: Review the example `CREATE TABLE` statement in the animation above. Carefully check the placement of commas, parentheses, and semicolon.

```
1 -- Write your CREATE TABLE statement here:
2
3
4
5 INSERT INTO Product (ID, Name, ProductType, OriginDate, Weight) VALUES
6   (100, 'Tricorder', 'TC', '2020-08-11', 2.4),
7   (200, 'Food replicator', 'FOD', '2020-09-21', 54.3)
```

```
  (200, 'Food replicator', 'FOD', '2020-09-21', 54.2),
  (300, 'Cloaking device', 'CD', '2019-02-04', 177.9);

SELECT *
FROM Product;
```

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Run Reset code

▶ View solution

ALTER TABLE statement

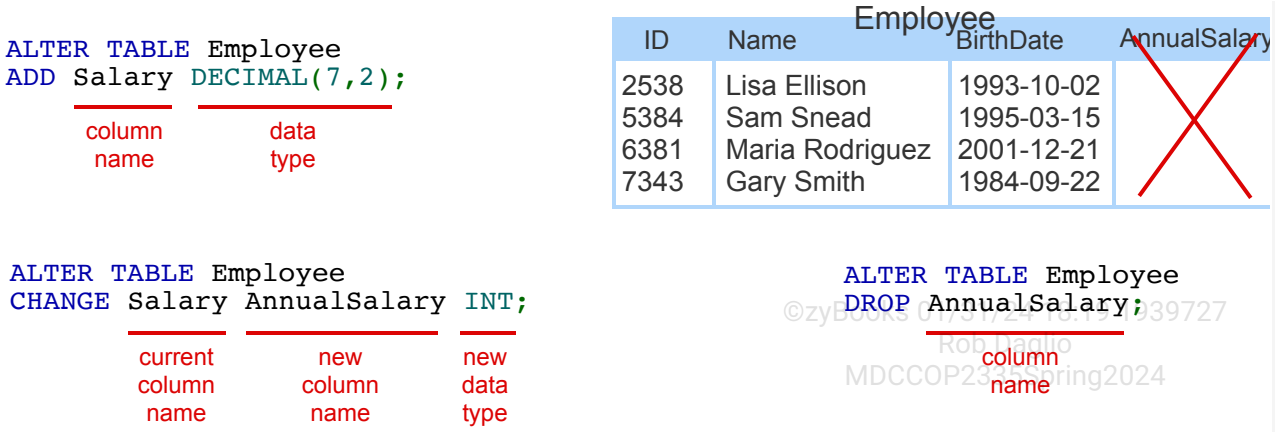
The ***ALTER TABLE*** statement adds, deletes, or modifies columns on an existing table. The ALTER TABLE statement specifies the table name followed by a clause that indicates what should be altered. The table below summarizes the three ALTER TABLE clauses.

Table 35.4.1: ALTER TABLE statement.

ALTER TABLE clause	Description	Syntax
ADD	Adds a column	`ALTER TABLE TableName` `ADD ColumnName DataType;`
CHANGE	Modifies a column	`ALTER TABLE TableName` `CHANGE CurrentColumnName NewColumnName NewDataType;`
DROP	Deletes a column	`ALTER TABLE TableName` `DROP ColumnName;`

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

PARTICIPATION ACTIVITY 35.4.8: Adding, modifying, and deleting columns.



©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Animation content:

Step 1. ALTER TABLE with the ADD clause adds a Salary column to Employee. Salary has 7 significant digits with 2 decimal places. The Employee table appears, with four rows and columns ID, Name, and BirthDate. An SQL query appears:
Begin SQL code:
ALTER TABLE Employee
ADD Salary DECIMAL(7,2);
End SQL code.
The caption "column name" appears under Salary. The caption "data type" appears under DECIMAL(7, 2). A new Salary column is added to the table.

Step 2. ALTER TABLE with the CHANGE clause changes the Salary column's name to AnnualSalary and data type to INT. A second SQL query appears:
Begin SQL code:
ALTER TABLE Employee
CHANGE Salary AnnualSalary INT;
End SQL code.
The caption "current column name" appears under Salary. The caption "new column name" appears under AnnualSalary. The caption "new data type" appears under INT. In the Employee table, the name of the Salary column changes to AnnualSalary.

Step 3. ALTER TABLE with the DROP clause deletes the AnnualSalary column. A third SQL query appears:
Begin SQL code:
ALTER TABLE Employee
DROP AnnualSalary;
End SQL code.
The caption "column name" appears under AnnualSalary. In the Employee table, a red X appears over the AnnualSalary column.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Animation captions:

1. ALTER TABLE with the ADD clause adds a Salary column to Employee. Salary has 7 significant digits with 2 decimal places.
2. ALTER TABLE with the CHANGE clause changes the Salary column's name to AnnualSalary and data type to INT.
3. ALTER TABLE with the DROP clause deletes the AnnualSalary column.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

PARTICIPATION ACTIVITY 35.4.9: ALTER TABLE statement.

1) Add a column called Description to the Department table.

```
ALTER TABLE Department
[ ]
VARCHAR(50);
```

Check Show answer

2) Rename column Description to ShortDesc.

```
ALTER TABLE Department
[ ]
VARCHAR(50);
```

Check Show answer

3) Change column ShortDesc to accept up to 80 characters.

```
ALTER TABLE Department
CHANGE
[ ]
;
```

Check Show answer

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

4) Delete the column ShortDesc.

```
ALTER [ ]
```

Check Show answer

CHALLENGE ACTIVITY 35.4.1: Creating, dropping, and altering tables.

539740.3879454.qx3zqy7

Start

Complete the following statement to create a table named Country.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Choose data types based on the following requirements:

- ISOCode3 stores the country's code, consisting of one to three letters.
- IndepDate stores the country's independence date.

```
/* Your code goes here */ (
    ISOCode3    /* Your code goes here */ ,
    IndepDate   /* Your code goes here */
);
```

Enter a statement to delete the above table.

```
/* Your code goes here */ ;
```

1 2 3

Check Next

Exploring further:

- MySQL CREATE TABLE syntax
- MySQL DROP TABLE syntax
- MySQL ALTER TABLE syntax

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

35.5 Data types

Data type categories

A ***data type*** is a named set of values from which column values are drawn. In relational databases, most data types fall into one of the following categories:

- ***Integer*** data types represent positive and negative integers. Several integer data types exist, varying by the number of bytes allocated for each value. Common integer data types include INT, implemented as 4 bytes of storage, and SMALLINT, implemented as 2 bytes.
- ***Decimal*** data types represent numbers with fractional values. Decimal data types vary by number of digits after the decimal point and maximum size. Common decimal data types include FLOAT and DECIMAL.
- ***Character*** data types represent textual characters. Common character data types include CHAR, a fixed string of characters, and VARCHAR, a string of variable length up to a specified maximum size.
- ***Date and time*** data types represent date, time, or both. Some date and time data types include a time zone or specify a time interval. Some date and time data types represent an interval rather than a point in time. Common date and time data types include DATE, TIME, DATETIME, and TIMESTAMP.
- ***Binary*** data types store data exactly as the data appears in memory or computer files, bit for bit. The database manages binary data as a series of zeros and ones. Common binary data types include BLOB, BINARY, VARBINARY, and IMAGE.
- ***Spatial*** data types store geometric information, such as lines, polygons, and map coordinates. Examples include POLYGON, POINT, and GEOMETRY. Spatial data types are relatively new and consequently vary greatly across database systems.
- ***Document*** data types contain textual data in a structured format such as XML or JSON.

Databases support additional data types for specialized uses. Ex: MONEY for currency values, BOOLEAN for true-false values, BIT for zeros and ones, and ENUM for a small, fixed set of alternative values.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Table 35.5.1: Example data types.

Category	Data type	Value
Integer	INT	-9281344
Decimal	FLOAT	3.1415
Character	VARCHAR	Chicago
Date and time	DATETIME	12/25/2020 10:35:00

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Binary	BLOB	1001011101 . . .
Spatial	POINT	(2.5, 33.44)
Document	XML	<menu> <selection> <name>Greek salad</name> <price>$13.90</price> <text>Cucumbers, tomatoes, onions, and feta cheese</text> </selection> <selection> <name>Turkey sandwich</name> <price>$9.00</price> <text>Turkey, lettuce, tomato on choice of bread</text> </selection> </menu>

©zyBooks 01/31/24 18:19 1939727 Rob Daglio MDCCOP2335Spring2024

PARTICIPATION ACTIVITY 35.5.1: Data types.

Match the value with the data type.

If unable to drag and drop, refresh the page.

09/12/1986 **3.14** **Nadia** **16** **00011011001**

	VARCHAR
	INT
	DATE
	BLOB
	FLOAT

©zyBooks 01/31/24 18:19 1939727 Rob Daglio MDCCOP2335Spring2024

Reset

PARTICIPATION ACTIVITY 35.5.2: Data types.

1) The CHAR data type represents a variable string of characters.
   - True
   - False

2) A binary data type stores data as an exact copy of computer memory.
   - True
   - False

3) Some date and time data types include time zone.
   - True
   - False

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

MySQL data types

All relational databases support integer, decimal, date and time, and character data types. Most databases allow integer and decimal numbers to be signed or unsigned. A ***signed*** number may be negative. An ***unsigned*** number cannot be negative.

Data types vary in storage requirements. Ex:

- Character data types use one or two bytes per character.
- Integer data types use a fixed number of bytes per number.
- Unsigned data types can store larger numbers than the signed version of the same data type.

To minimize table size, the data type with the smallest storage requirements should be used. Ex: Any integer data type can store integers that range from -100 to 100. However, TINYINT requires only one byte per number and is the best choice for this range.

Common MySQL data types appear in the table below. For a complete list of MySQL data types, see the link in Exploring Further, below.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Table 35.5.2: Common MySQL data types.

Category	Example	Data type	Storage	Notes
Integer	34 and -739448	TINYINT	1 byte	Signed range: -128 to 127 Unsigned range: 0 to 255

Category	Example	Data type	Storage	Notes
		SMALLINT	2 bytes	Signed range: -32,768 to 32,767 Unsigned range: 0 to 65,535
		MEDIUMINT	3 bytes	Signed range: -8,388,608 to 8,388,607 Unsigned range: 0 to 16,777,215
		INTEGER or INT	4 bytes	Signed range: -2,147,483,648 to 2,147,483,647 Unsigned range: 0 to 4,294,967,295
		BIGINT	8 bytes	Signed range: -2^{63} to 2^{63} -1 Unsigned range: 0 to 2^{64} -1
Decimal	123.4 and -54.29685	DECIMAL(M,D)	Varies depending on *M* and *D*	Exact decimal number where *M* = number of significant digits, *D* = number of digits after decimal point
		FLOAT	4 bytes	Approximate decimal numbers with range: -3.4E+38 to 3.4E+38
		DOUBLE	8 bytes	Approximate decimal numbers with range: -1.8E+308 to 1.8E+308
Date and time	'1776-07-04 13:45:22'	DATE	3 bytes	Format: YYYY-MM-DD. Range: '1000-01-01' to '9999-12-31'
		TIME	3 bytes	Format: hh:mm:ss
		DATETIME	5 bytes	Format: YYYY-MM-DD hh:mm:ss. Range: '1000-01-01 00:00:00' to '9999-12-31 23:59:59'.
Character	'string'	CHAR(N)	*N* bytes	Fixed-length string of length *N*; 0 ≤ *N* ≤ 255

©zyBooks 01/31/24 18:19 1939727 Rob Daglio MDCCOP2335Spring2024

Category	Example	Data type	Storage	Notes
		VARCHAR(N)	Length of characters + 1 bytes	Variable-length string with maximum *N* characters; 0 ≤ *N* ≤ 65,535
		TEXT	Length of characters + 2 bytes	Variable-length string with maximum 65,535 characters

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

PARTICIPATION ACTIVITY 35.5.3: Common MySQL data types.

Choose the best data type for each scenario with the minimum storage requirements.

1) Storing a city's population, which ranges from a dozen to 24 million people.
   - unsigned MEDIUMINT
   - unsigned INTEGER
   - unsigned BIGINT

2) Storing the annual gain or loss in a city's population, which ranges from -1 million to 1 million.
   - signed SMALLINT
   - unsigned MEDIUMINT
   - signed MEDIUMINT

3) Storing the price of an item that ranges from a few dollars to a few hundred dollars.
   - FLOAT
   - DECIMAL(2,5)
   - DECIMAL(5,2)

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

4) Storing the date and time an item is purchased.

- DATE
- TIME
- DATETIME

5) Storing a student's assigned letter grade, like A or D.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

- TINYINT
- CHAR(1)
- VARCHAR(1)

6) Storing a student's email address.

- VARCHAR(100)
- VARCHAR(10)
- CHAR(100)

PARTICIPATION ACTIVITY 35.5.4: Create a Product table with MySQL data types.

Write the SQL to create a Product table with the following columns:

- *ID* - Integer with range 0 to 65,535
- *Name* - Variable-length string with maximum 40 characters
- *ProductType* - Fixed-length string with 3 characters
- *OriginDateTime* - Year, month, day, and time
- *Weight* - Decimal number with variable precision and 4 bytes of storage

Place your CREATE TABLE statement before the INSERT and SELECT statements. Run your solution and verify the result table contains the three inserted rows.

```
1  -- Write your CREATE TABLE statement here:
2
3
4
5  INSERT INTO Product (ID, Name, ProductType, OriginDateTime, Weight) VALU
6    (100, 'Tricorder', 'COM', '2020-08-11 11:30:00', 2.4172),
7    (200, 'Food replicator', 'FOD', '2020-09-21 14:00:00', 54.02),
8    (300, 'Cloaking device', 'SPA', '2019-02-04 07:35:00', 19257.9);
9
10 SELECT *
11 FROM Product;
```

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Run Reset code

▶ View solution

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Exploring further:

- MySQL data types

35.6 Selecting rows

Operators

An ***operator*** is a symbol that computes a value from one or more other values, called ***operands***:

- Arithmetic operators compute numeric values from numeric operands.
- Comparison operators compute logical values TRUE or FALSE. Operands may be numeric, character, and other data types.
- Logical operators compute logical values from logical operands.

A ***unary*** operator has one operand. A ***binary*** operator has two operands. Most operators are binary. The logical operator NOT is unary. The arithmetic operator - is either unary or binary.

Operators may return NULL when either operand is NULL. NULL-valued operations are discussed elsewhere in this material.

Table 35.6.1: Common operators.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Type	Operator	Description	Example	Value
Arithmetic	+	Adds two numeric values	`4 + 3`	7
	– (unary)	Reverses the sign of one	`-(-2)`	2

		numeric value		
	– (binary)	Subtracts one numeric value from another	`11 - 5`	`6`
	*	Multiplies two numeric values	`3 * 5`	`15`
	/	Divides one numeric value by another	`4 / 2`	`2`
	% (modulo)	Divides one numeric value by another and returns the integer remainder	`5 % 2`	`1`
	^	Raises one numeric value to the power of another	`5^2`	`25`
Comparison	=	Compares two values for equality	`1 = 2`	`FALSE`
	!= <>	Compares two values for inequality	`1 != 2` `1 <> 2`	`TRUE`
	<	Compares two values with <	`2 < 2`	`FALSE`
	<=	Compares two values with ≤	`2 <= 2`	`TRUE`
	>	Compares two values with >	`'2019-08-13' > '2021-08-13'`	`FALSE`
	>=	Compares two values with ≥	`'apple' >= 'banana'`	`FALSE`

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Logical	`AND`	Returns TRUE only when both values are TRUE	`TRUE AND FALSE`	`FALSE`
	`OR`	Returns FALSE only when both values are FALSE	`TRUE OR FALSE`	`TRUE`
	`NOT`	Reverses a logical value	`NOT FALSE`	`TRUE`

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

PARTICIPATION ACTIVITY 35.6.1: Operators.

Match the operator with the description.

If unable to drag and drop, refresh the page.

NOT **-** **AND** **%** **!=**

	Binary arithmetic operator
	Either unary or binary arithmetic operator
	Binary comparison operator
	Unary logical operator
	Binary logical operator

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Reset

Expressions

An ***expression*** is a string of operators, operands, and parentheses that evaluates to a single value. Operands may be column names or fixed values. The value of an expression may be any data type. Ex:

`Salary > 34000 AND Department = 'Marketing'` is an expression with a logical value.

A simple expression may consist of a single column name or a fixed value. Ex: The column `EmployeeName` and the fixed value `'Maria'` are expressions with a character data type.

When an expression is evaluated, column names are replaced with column values for a specific row. Consequently, an expression containing column names may have different values for different rows.

The order of operator evaluation may affect the value of an expression. Operators in an expression are evaluated in the order of ***operator precedence***, shown in the table below. Operators of the same precedence are evaluated from left to right. Regardless of operator precedence, expressions enclosed in parentheses are evaluated before any operators outside the parentheses are applied.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Expressions may evaluate to NULL. NULL-valued expressions are discussed elsewhere in this material.

Table 35.6.2: Operator precedence.

Precedence	Operators
1	- (unary)
2	^
3	* / %
4	+ - (binary)
5	= != < > <= >=
6	NOT
7	AND
8	OR

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

PARTICIPATION ACTIVITY 35.6.2: Evaluating expressions.

Expression: Status = 'Platinum' AND (Quantity * UnitPrice + ShippingPrice) > 100.00

'Platinum' = 'Platinum' AND	(4 * 15.00 + 7.50)	> 100.00	
'Platinum' = 'Platinum' AND	(60.00 + 7.50)	> 100.00	
'Platinum' = 'Platinum' AND	67.50	> 100.00	
TRUE AND	67.50	> 100.00	
TRUE AND	FALSE		
Value:	FALSE		

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Animation content:

Static figure:
An expression appears with evaluation steps on subsequent lines:
Status = 'Platinum' AND (Quantity * UnitPrice + ShippingPrice) > 100.00
'Platinum' = 'Platinum' AND (4 * 15.00 + 7.50) > 100.00
'Platinum' = 'Platinum' AND (60.00 + 7.50) > 100.00
'Platinum' = 'Platinum' AND 67.50 > 100.00
TRUE AND 67.50 > 100.00
TRUE AND FALSE
FALSE

Step 1: The expression includes column names Status, Quantity, UnitPrice, and ShippingPrice. The initial expression appears:
Status = 'Platinum' AND (Quantity * UnitPrice + ShippingPrice) greater than 100.00

Step 2: The expression is evaluated for a specific row. Column names are replaced with column values for the row. The first evaluation step appears below the initial expression:
'Platinum' = 'Platinum' AND (4 * 15.00 + 7.50) greater than 100.00

Step 3:Operators within parentheses have the highest precedence. * has higher precedence than + and is evaluated first. The next evaluation step appears below the prior step:
'Platinum' = 'Platinum' AND (60.00 + 7.50) greater than 100.00

Step 4: + is the only remaining operator inside the parentheses, and is evaluated next. The next evaluation step appears below the prior step:
'Platinum' = 'Platinum' AND 67.50 greater than 100.00

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Step 5: = has higher precedence than AND. = has the same precedence as > but appears to the left. So = is evaluated next. The next evaluation step appears below the prior step:
TRUE AND 67.50 greater than 100.00

Step 6: "greater than" has higher precedence than AND, and is evaluated next. The next evaluation

step appears below the prior step:
TRUE AND FALSE

Step 7: The expression evaluates to FALSE for the row. FALSE appears under the prior step.

Animation captions:

1. The expression includes column names Status, Quantity, UnitPrice, and ShippingPrice.
2. The expression is evaluated for a specific row. Column names are replaced with column values for the row.
3. Operators within parentheses have the highest precedence. * has higher precedence than + and is evaluated first.
4. + is the only remaining operator inside the parentheses, and is evaluated next.
5. = has higher precedence than AND. = has the same precedence as > but appears to the left. So = is evaluated next.
6. > has higher precedence than AND, and is evaluated next.
7. The expression evaluates to FALSE for the row.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

PARTICIPATION ACTIVITY 35.6.3: Expressions.

What is the value of the following expressions?

1) `7 + 3 * 2`

- 13
- 20

2) `(7 + 3) * 2`

- 13
- 20

3) `(8 % 3 + 10 > 15) AND TRUE`

- TRUE
- FALSE

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

4) `(Age >= 13 AND Age <= 18) OR Military = 'Army'`
where Age = 8 and Military = 'Army'

- TRUE
- FALSE

SELECT statement

The SELECT statement selects rows from a table. The statement has a ***SELECT*** clause and a ***FROM*** clause. The FROM clause specifies the table from which rows are selected. The SELECT clause specifies one or more expressions, separated by commas, that determine what values are returned for each row.

In many queries, each expression in the SELECT clause is a simple column name. To select all columns, the expressions can be replaced with a single asterisk.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Figure 35.6.1: SELECT statement.

```
-- SELECT with expressions
SELECT Expression1, Expression2,
...
FROM TableName;

-- SELECT with column names
SELECT Column1, Column2, ...
FROM TableName;

-- SELECT with asterisk
SELECT *
FROM TableName;
```

The SELECT statement returns a set of rows, called the ***result table***.

PARTICIPATION ACTIVITY 35.6.4: Selecting from table CountryLanguage.

CountryLanguage

CountryCode	Language	IsOfficial	Percentage
ABW	Dutch	T	5.3
AFG	Balochi	F	0.9
AGO	Kongo	F	13.2
ALB	Albanian	T	97.9

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

```
SELECT *
FROM CountryLanguage;
```

CountryCode	Language	IsOfficial	Percentage
ABW	Dutch	T	5.3
AFG	Balochi	F	0.9
AGO	Kongo	F	13.2
ALB	Albanian	T	97.9

```
SELECT CountryCode, Language
FROM CountryLanguage;
```

CountryCode	Language
ABW	Dutch
AFG	Balochi
AGO	Kongo
ALB	Albanian

Animation content:

Step 1: The CountryLanguage table contains 4 rows with columns CountryCode, Language, IsOfficial, and Percentage. The CountryLanguage table appears.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Step 2: The SELECT statement selects all columns using *. All rows and columns appear in the result table. An SQL statement appears:
Begin SQL code:
SELECT *
FROM Country Language;
End SQL code.
An unnamed table appears, with the same columns as CountryLanguage. All data in CountryLanguage is duplicated and moved into the unnamed table.

Step 3: The SELECT statement selects only columns CountryCode and Language, so only two columns appear in the result table. An SQL statement appears:
Begin SQL code:
SELECT CountryCode, Language
FROM CountryLanguage;
End SQL code.
A new table appears, with columns CountryCode and Language. The data in columns CountryCode and Language of table CountryLanguage is duplicated and moved into the new table.

Animation captions:

1. The CountryLanguage table contains 4 rows with columns CountryCode, Language, IsOfficial, and Percentage.
2. The SELECT statement selects all columns using *. All rows and columns appear in the result table.
3. The SELECT statement selects only columns CountryCode and Language, so only two columns appear in the result table.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

PARTICIPATION ACTIVITY 35.6.5: SELECT with columns.

Refer to the following table.

City

ID	Name	CountryCode	District	Population
1	Kabul	AFG	Kabol	1780000

ID	Name	CountryCode	District	Population
2	Qandahar	AFG	Qandahar	237500
3	Amsterdam	NLD	Noord-Holland	731200
4	Rotterdam	NLD	Zuid-Holland	593321
5	Sydney	AUS	New South Wales	3276207

1) Select all rows and only the Name and District columns.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

```
SELECT
FROM City;
```

Check Show answer

2) Select all rows and columns.

```
SELECT
FROM City;
```

Check Show answer

3) Select all rows and columns except ID in order of columns shown.

```
SELECT
FROM City;
```

Check Show answer

PARTICIPATION ACTIVITY 35.6.6: SELECT with expressions.

Refer to the following table.

Customer

ID	Name	Balance	Payment
193	Chen	2100	300
584	Ravindran	5000	250
231	Bolt	300	10
608	Gomez	950	125

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

1) What values are returned?

```
SELECT Balance + Payment
FROM Customer;
```

- 2400, 5250, 310, 1075
- 2400
- 2100, 5000, 300, 950

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

2) What values are returned?

```
SELECT 2 * (Balance - Payment)
FROM Customer;
```

- 3900, 9750, 590, 1775
- 3600
- 3600, 9500, 580, 1650

Long tables

*Some tables may contain thousands or millions of rows, and selecting all rows can take a long time. MySQL has a **LIMIT** clause that limits the number of rows returned by a SELECT statement. Ex: The SELECT statement below returns only the first 100 rows from the City table.*

```
SELECT *
FROM City
LIMIT 100;
```

WHERE clause

An expression may return a value of any data type. A ***condition*** is an expression that evaluates to a logical value.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

A SELECT statement has an optional ***WHERE*** clause that specifies a condition for selecting rows. A row is selected when the condition is TRUE for the row values. A row is omitted when the condition is either FALSE or NULL.

The WHERE clause follows the FROM clause. When a SELECT statement has no WHERE clause, all rows are selected.

Figure 35.6.2: WHERE clause.

```
SELECT Expression1, Expression2,
...
FROM TableName
WHERE Condition;
```

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

PARTICIPATION ACTIVITY 35.6.7: WHERE clause.

CountryLanguage

CountryCode	Language	IsOfficial	Percentage
ABW	Dutch	T	5.3
AFG	Balochi	F	0.9
AGO	Kongo	F	13.2
ALB	Albanian	T	97.9

```
SELECT CountryCode, Language
FROM CountryLanguage
WHERE Percentage > 0.0
 AND Percentage < 10.0;
```

CountryCode	Language
ABW	Dutch
AFG	Balochi

```
SELECT CountryCode, Language
FROM CountryLanguage
WHERE Percentage < 5.0
 OR Percentage > 90.0;
```

CountryCode	Language
AFG	Balochi
ALB	Albanian

Animation content:

Static figure:
The CountryLanguage table appears, with columns CountryCode, Language, IsOfficial, and Percentage. CountryLanguage has four rows:
ABW, Dutch, T, 5.3
AFG, Balochi, F, 0.9
AGO, Kongo, F, 13.2
ALB, Albanian, T, 97.9

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

A query appears below the table:
Begin SQL code:

```
SELECT CountryCode, Language
FROM CountryLanguage
WHERE Percentage > 0.0
AND Percentage < 10.0;
End SQL code.
```

The result table appears next to the query, with columns CountryCode and Language. The values for these columns in the first two rows of CountryLanguage appear in the result table.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

A second query appears below the first query:

```
Begin SQL code:
SELECT CountryCode, Language
FROM CountryLanguage
WHERE Percentage < 5.0
OR Percentage > 90.0;
End SQL code.
```

A second result table appears next to the second query, with columns CountryCode and Language. The values for these columns in rows two and four of CountryLanguage appear in the result table.

Step 1: The statement selects rows with percentage between 0.0 and 10.0. Two rows are returned. The first query appears. The expression (Percentage greater than 0.0 and Percentage less than 10.0) is evaluated for each row of CountryLanguage. TRUE appears next to the first two rows. FALSE appears next to the last two rows. The first result table appears.

Step 2: The statement selects rows with percentage less than 5.0 or percentage greater than 90.0. Two rows are returned. The second query appears. The expression (Percentage less than 5.0 or Percentage greater than 90.0) is evaluated for each row of CountryLanguage. TRUE appears next to rows two and four. FALSE appears next to rows one and three. The second result table appears.

Animation captions:

1. The statement selects rows with percentage between 0.0 and 10.0. Two rows are returned.
2. The statement selects rows with percentage < 5.0 or percentage > 90.0. Two rows are returned.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

PARTICIPATION ACTIVITY 35.6.8: WHERE clause.

Refer to the City table.

City

ID	Name	CountryCode	District	Population

ID	Name	CountryCode	District	Population
207	Rio de Janeiro	BRA	Rio de Janeiro	5598953
462	Manchester	GBR	England	430000
554	Santiago de Chile	CHL	Santiago	4703954
654	Barcelona	ESP	Katalonia	1503451
826	Valencia	PHL	Northern Mindanao	147924
1025	Delhi	IND	Delhi	7206704

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

1) What cities are selected?

```
SELECT *
FROM City
WHERE Population < 150000;
```

- All cities
- Valencia and Manchester
- Valencia

2) What districts are selected?

```
SELECT District
FROM City
WHERE Name = 'Rio de Janeiro';
```

- No districts
- Rio de Janeiro
- Rio de Janeiro and Katalonia

3) What cities are selected?

```
SELECT Name
FROM City
WHERE CountryCode != 'CHL';
```

- No cities
- All cities
- All cities except Santiago de Chile

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

4) What names are selected?

```
SELECT Name
FROM City
WHERE NOT ID < 2000;
```

- No names
- All names
- Valencia

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

5) What cities are selected?

```
SELECT *
FROM City
WHERE ID <= 1000
  AND ID >= 600;
```

- All cities
- Barcelona and Valencia
- Delhi

6) What cities are selected?

```
SELECT *
FROM City
WHERE ID < 300
   OR Population > 7500000;
```

- All cities
- Delhi and Rio de Janeiro
- Rio de Janeiro

7) What cities are selected?

```
SELECT *
FROM City
WHERE (Population >= 7000000
AND Population <= 8000000)
   OR ID < 500;
```

- All cities
- Rio de Janeiro and Manchester
- Rio de Janeiro, Manchester, and Delhi

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

PARTICIPATION ACTIVITY 35.6.9: Select movies with logical operators.

The given SQL creates a Movie table and inserts some movies. The SELECT statement selects all movies released before January 1, 2000.

Modify the SELECT statement to select the title and release date of PG-13 movies that are released after January 1, 2008.

Run your solution and verify the result table shows just the titles and release dates for *The Dark Knight* and *Crazy Rich Asians*.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

```
CREATE TABLE Movie (
  ID INT AUTO_INCREMENT,
  Title VARCHAR(100),
  Rating CHAR(5) CHECK (Rating IN ('G', 'PG', 'PG-13', 'R')),
  ReleaseDate DATE,
  PRIMARY KEY (ID)
);

INSERT INTO Movie (Title, Rating, ReleaseDate) VALUES
  ('Casablanca', 'PG', '1943-01-23'),
  ('Bridget Jones\'s Diary', 'PG-13', '2001-04-13'),
  ('The Dark Knight', 'PG-13', '2008-07-18'),
  ('Hidden Figures', 'PG', '2017-01-06'),
  ('Toy Story', 'G', '1995-11-22'),
  ('Rocky', 'PG', '1976-11-21'),
  ('Crazy Rich Asians', 'PG-13', '2018-08-15');

-- Modify the SELECT statement:
SELECT *
FROM Movie
WHERE ReleaseDate < '2000-01-01';

```

Run Reset code

▶ View solution

CHALLENGE ACTIVITY 35.6.1: Selecting rows.

539740.3879454.qx3zqy7

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Start

Country

Area	ContinentCode	IndepYear
287100	SA	1810
95900	SA	1809
88800	EU	1877

What columns are returned by the given statements?

```
SELECT *
FROM Country;
```

- [ ] Area
- [ ] ContinentCode
- [ ] IndepYear

```
SELECT IndepYear
FROM Country;
```

- [ ] Area
- [ ] ContinentCode
- [ ] IndepYear

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

1 2 3 4 5

Check Next

Exploring further:

- MySQL SELECT syntax
- MySQL operators

35.7 Null values

NULL

NULL is a special value that represents either unknown or inapplicable data. NULL is not the same as zero for numeric data types or blanks for character data types. Ex: A zero bonus indicates an employee can, but has not, earned a bonus. A zero bonus is known and applicable, and should not be represented as NULL.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

PARTICIPATION ACTIVITY 35.7.1: NULL values in the Compensation table.

NOT NULL

Compensation

ID	Name	BirthDate	Salary	Department	Bonus

2538	Lisa Ellison	October 2, 1993	45000	Engineering	NULL
5384	Sam Snead	NULL	32000	Sales	1000
6381	Maria Rodriguez	December 21, 2001	95000	Sales	3000

Animation content:

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Static figure:
The Compensation table appears, with columns ID, Name, BirthDate, Salary, Department, and Bonus. Compensation has three rows:
2538, Lisa Ellison, October 2 1993, 45000, Engineering, NULL
5384, Sam Snead, NULL, 32000, Sales, 1000
6381, Maria Rodriguez, December 21 2001, 95000, Sales, 3000
A caption NOT NULL appears above the ID column. The two NULL values in Compensation are highlighted.

Step 1: A NULL in the BirthDate column means "unknown", since all employees have a birth date. The NULL in row two of column BirthDate is highlighted.

Step 2: If Engineering employees are not paid a bonus, Lisa Ellison's NULL bonus means "inapplicable". The NULL in row one of column Bonus is highlighted.

Step 3: The ID column identifies employees and must contain valid data. The column is designated NOT NULL and cannot accept a NULL value or missing data. The caption NOT NULL appears above the ID column.

Animation captions:

1. A NULL in the BirthDate column means "unknown", since all employees have a birth date.
2. If Engineering employees are not paid a bonus, Lisa Ellison's NULL bonus means "inapplicable".
3. The ID column identifies employees and must contain valid data. The column is designated NOT NULL and cannot accept a NULL value or missing data.

PARTICIPATION ACTIVITY 35.7.2: NULL values.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Refer to the table below.

Compensation

ID	Name	Department	Salary	Bonus
2538	Lisa Ellison	Engineering	45000	0
5384	Sam Snead	Sales	30500	1000

ID	Name	Department	Salary	Bonus
6381	NULL	Sales	92300	3000

1) What does a NULL in the Name column represent?

- Unknown
- Inapplicable
- Either unknown or inapplicable

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

2) In the Bonus column of the Lisa Ellison row, what does the zero represent?

- Lisa Ellison's bonus is unknown.
- Lisa Ellison is not eligible for a bonus.
- Lisa Ellison has earned no bonus.

NOT NULL constraint

By default, columns may contain NULL values. In some cases, however, columns should never contain NULL. Ex: If a business requires that a name is specified for all employees, the Name column of an Employee table should not contain NULL.

The ***NOT NULL*** constraint prevents a column from having a NULL value. Statements that insert NULL, or update a value to NULL, are automatically rejected. NOT NULL follows the column name and data type in a CREATE TABLE statement.

PARTICIPATION ACTIVITY 35.7.3: NOT NULL constraint.

```
CREATE TABLE Employee (
   ID          SMALLINT UNSIGNED,
   Name        VARCHAR(60) NOT NULL,
   BirthDate   DATE,
   Salary      DECIMAL(7,2)
);
```

Employee

ID	Name	BirthDate	Salary
6381	Maria Rodriguez	NULL	92300
~~2538~~	~~NULL~~	~~1990-12-03~~	~~423.00~~

Animation content:

Static figure:
This query appears:
Begin SQL code:

```
CREATE TABLE Employee (
ID SMALLINT UNSIGNED,
Name VARCHAR(60) NOT NULL,
BirthDate DATE,
Salary DECIMAL(7,2)
);
```

End SQL code.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

The Employee table appears, with columns ID, Name, BirthDate, and Salary. Employee has two rows:
6381, Maria Rodriguez, NULL, 92300
2538, NULL, 1990-03, 423.00
The second row is struck through with a red line.

Step 1: The BirthDate column allows NULL values by default. The query clause BirthDate DATE is highlighted. The first row of Employee appears.

Step 2: The NOT NULL constraint prevents Name from being NULL when inserting a new row into Employee. The query keywords NOT NULL are highlighted. The second row of Employee appears. NULL is highlighted in this row. The row is struck through with a red line.

Animation captions:

1. The BirthDate column allows NULL values by default.
2. The NOT NULL constraint prevents Name from being NULL when inserting a new row into Employee.

PARTICIPATION ACTIVITY 35.7.4: NOT NULL constraint.

Refer to the statement below.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

```
CREATE TABLE Department (
   Code      TINYINT UNSIGNED NOT NULL,
   Name      VARCHAR(20),
   ManagerID SMALLINT
);
```

1) Which columns may contain NULL values?

- Code
- Code and Name
- Name and ManagerID

2) Which alteration to the CREATE TABLE statement prevents ManagerID from being NULL?

- `ManagerID NOT NULL SMALLINT`
- `ManagerID NOT NULL`
- `ManagerID SMALLINT NOT NULL`

3) What happens when a user attempts to insert a new department without a Code value?

- The database accepts the insert and assigns Code with zero.
- The database accepts the insert and assigns Code with NULL.
- The database rejects the insert.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

NULL arithmetic and comparisons

When arithmetic or comparison operators have one or more NULL operands, the result is NULL. When a WHERE clause evaluates to NULL for values in a row, the row is not selected.

PARTICIPATION ACTIVITY 35.7.5: NULL arithmetic and comparisons.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Compensation

ID	Name	BirthDate	Salary	Department	Bonus
2538	Lisa Ellison	October 2, 1993	45000	Engineering	NULL
5384	Sam Snead	NULL	32000	Sales	1000
6381	Maria Rodriguez	December 21, 2001	95000	Sales	3000

```
SELECT Name
FROM Compensation
WHERE (Salary + Bonus) > 30000;
```

NULL

Result

Name
Sam Snead
Maria Rodriguez

```
SELECT Name
FROM Compensation
WHERE BirthDate = NULL;
```

NULL

Result

Name
No rows returned

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Animation content:

Static figure:
The Compensation table appears, with columns ID, Name, BirthDate, Salary, Department, and Bonus. Compensation has three rows:
2538, Lisa Ellison, October 2 1993, 45000, Engineering, NULL
5384, Sam Snead, NULL, 32000, Sales, 1000
6381, Maria Rodriguez, December 21 2001, 95000, Sales, 3000

A query appears:
Begin SQL code:
SELECT Name
FROM Compensation
WHERE (Salary + Bonus) > 3000;
End SQL code.
NULL appears under the expression in the WHERE clause. The result appears next to this query, with column Name and two rows:
Sam Snead
Maria Rodriguez

A second query appears.
Begin SQL code:
SELECT Name
FROM Compensation
WHERE BirthDate = NULL;
End SQL code.
NULL appears under the expression in the WHERE clause. The result appears next to this query, with column Name and no rows.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio

Step 1: Lisa Ellison's Bonus is NULL. As a result, Salary + Bonus is NULL and (Salary + Bonus) > 30000 is NULL. The first query appears. The WHERE clause evaluates to NULL for the first row of Compensation.

Step 2: The SELECT statement does not select a name from a row when the WHERE clause is NULL, so Lisa Ellison is not selected. The first result table appears without Lisa Ellison.

Step 3: The = comparison operator returns NULL when either or both operands are NULL, so the WHERE clause evaluates to NULL for Sam Snead. The second query appears. NULL appears under the WHERE clause. The second result table appears with no rows.

Animation captions:

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

1. Lisa Ellison's Bonus is NULL. As a result, Salary + Bonus is NULL and (Salary + Bonus) > 30000 is NULL.
2. The SELECT statement does not select a name from a row when the WHERE clause is NULL, so Lisa Ellison is not selected.
3. The = comparison operator returns NULL when either or both operands are NULL, so the WHERE clause evaluates to NULL for Sam Snead.

PARTICIPATION ACTIVITY 35.7.6: NULL arithmetic and comparisons.

Refer to the table below.

Compensation

ID	Name	Salary	Bonus
2538	Lisa Ellison	45000	NULL
5348	Sam Snead	32000	32000
6381	Maria Rodriguez	95000	98000
8820	Jiho Chen	NULL	NULL

What name is selected by each statement?

1)
```
SELECT Name
FROM Compensation
WHERE Salary = Bonus;
```

Check Show answer

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

2)
```
SELECT Name
FROM Compensation
WHERE (Salary / Bonus) <
1.0;
```

Check Show answer

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

IS NULL operator

Since comparison operators return NULL when either operand is NULL, comparison operators cannot be used to select NULL values. Ex: `SELECT * FROM Employee WHERE Salary = NULL;` never returns any rows, because the WHERE clause is always NULL.

Instead, the ***IS NULL*** and ***IS NOT NULL*** operators must be used to select NULL values. `Value IS NULL` returns TRUE when the value is NULL. `Value IS NOT NULL` returns TRUE when the value is not NULL.

PARTICIPATION ACTIVITY 35.7.7: Selecting NULL values.

Country

Code	Name	HeadOfState	IndepYear	Population
ABW	Aruba	Beatrix	NULL	103000
AIA	Anguilla	Charles III	1920	NULL
AFG	Afghanistan	Mohammad Omar	1919	22720000
AGO	Angola	Jose dos Santos	1975	12878000

```
SELECT *
FROM Country
WHERE IndepYear IS NULL;
```

Code	Name	HeadOfState	IndepYear	Population
ABW	Aruba	Beatrix	NULL	103000

```
SELECT *
FROM Country
WHERE Population IS NOT NULL;
```

Code	Name	HeadOfState	IndepYear	Population
ABW	Aruba	Beatrix	NULL	103000
AFG	Afghanistan	Mohammad Omar	1919	22720000
AGO	Angola	Jose dos Santos	1975	12878000

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Animation content:

Static figure:
The Country table appears with columns Code, Name, HeadOfState, IndepYear, and Population. Country has four rows:
ABW, Aruba, Beatrix, NULL, 103000
AIA, Anguilla, Charles III, 1920, NULL
AFG, Afghanistan, Mohammad Omar, 1919, 22720000
AGO, Angola, Jose dos Santos, 1975, 12878000

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

A query appears:
Begin SQL code:
SELECT *
FROM Country
WHERE IndepYear IS NULL;
End SQL code.
The result table appears next to the query, with the same columns as Country. The result has one row:
ABW, Aruba, Beatrix, NULL, 103000

A second query appears:
Begin SQL code:
SELECT *
FROM Country
WHERE Population IS NOT NULL;
End SQL code.
A second result table appears next to the second query, with the same columns as Country. The result has three rows:
ABW, Aruba, Beatrix, NULL, 103000
AFG, Afghanistan, Mohammad Omar, 1919, 22720000
AGO, Angola, Jose dos Santos, 1975, 12878000

Step 1: The NULL in column IndepYear represents inapplicable data - a country has not achieved independence. The NULL in column Population represents unknown data. The Country table appears.

Step 2: Selecting rows where IndepYear IS NULL returns in one row. The first query appears. The first result table appears.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Step 3: Selecting rows where Population IS NOT NULL returns three rows. The second query appears. The second result table appears.

Animation captions:

1. The NULL in column IndepYear represents inapplicable data - a country has not achieved independence. The NULL in column Population represents unknown data.

2. Selecting rows where IndepYear IS NULL returns in one row.

PARTICIPATION ACTIVITY 35.7.8: Selecting NULL values.

Refer to the table below.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Country

Code	Name	HeadOfState	IndepYear	Population
ABW	Aruba	Beatrix	NULL	103000
AIA	Anguilla	Charles III	1920	NULL
AFG	Afghanistan	Mohammad Omar	1919	22720000
AGO	Angola	Jose dos Santos	1975	12878000

1) How many rows are returned?

```
SELECT *
FROM Country
WHERE Population = NULL;
```

- 0
- 1
- 3

2) How many rows are returned?

```
SELECT *
FROM Country
WHERE Population IS NULL;
```

- 0
- 1
- 3

3) What is missing to select all rows except Aruba?

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

```
SELECT *
FROM Country
WHERE IndepYear ______;
```

- != NULL
- IS NULL
- IS NOT NULL

PARTICIPATION ACTIVITY 35.7.9: Select songs with NULL values.

The given SQL creates a Song table and inserts some songs. Some values are NULL. A SELECT statement selects songs that have a NULL Artist value.

Press the Run button to produce a result table. Verify 'Need You Now' and 'That's The Way Love Goes' are the only two songs selected.

©zyBooks 01/31/24 18:19 1939727 Rob Daglio MDCCOP2335Spring2024

Modify the SELECT statement to only select songs that have a NULL Title value. Then run your solution, and verify the new query returns only songs by Taylor Swift and Nirvana.

```
CREATE TABLE Song (
  ID INT,
  Title VARCHAR(60),
  Artist VARCHAR(60),
  ReleaseYear INT
);

INSERT INTO Song VALUES
  (100, 'Hey Jude', 'Beatles', 1968),
  (200, NULL, 'Taylor Swift', 2008),
  (300, 'Need You Now', NULL, 2011),
  (400, 'That\'s The Way Love Goes', NULL, 1993),
  (500, NULL, 'Nirvana', 1991);

-- Modify the SELECT statement:
SELECT *
FROM Song
WHERE Artist IS NULL;
```

Run Reset code

▶ View solution

NULL logic

In traditional mathematical logic, expressions are always TRUE or FALSE. When NULL is present, however, a logical expression may be either TRUE, FALSE, or NULL. NULL indicates the value of a logical expression is uncertain. Ex:

©zyBooks 01/31/24 18:19 1939727 Rob Daglio MDCCOP2335Spring2024

- TRUE AND TRUE is TRUE.
- TRUE AND FALSE is FALSE.
- TRUE AND NULL is NULL.

The value of logical expressions containing NULL operands is defined in ***truth tables***.

Figure 35.7.1: MySQL truth tables.

x	y	x AND y	x OR y
TRUE	NULL	NULL	TRUE
NULL	TRUE		
FALSE	NULL	FALSE	NULL
NULL	FALSE		
NULL	NULL	NULL	NULL

x	NOT x
NULL	NULL

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

MySQL does not have a special data type for logical values. Internally, MySQL represents FALSE as 0 and TRUE as 1. In query results, FALSE and TRUE are also displayed as 0 and 1.

Since null logic is not standardized in mathematics, implementation details vary. Ex: Oracle Database displays a NULL logical expression as UNKNOWN.

PARTICIPATION ACTIVITY 35.7.10: NULL logic.

Refer to the Compensation table below.

Compensation

ID	Name	Salary	Bonus
2538	Lisa Ellison	115000	NULL
5348	Sam Snead	35000	55000
6381	Maria Rodriguez	95000	3000

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

In MySQL, what names are selected by the following queries?

1)
```
SELECT Name
FROM Compensation
WHERE Salary > 30000 OR Bonus >
1000;
```

- Lisa Ellison
- Sam Snead and Maria Rodriguez
- Lisa Ellison, Sam Snead, and Maria Rodriguez

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

2)
```
SELECT Name
FROM Compensation
WHERE Salary > 30000 AND BONUS
> 1000;
```

- Lisa Ellison
- Sam Snead and Maria Rodriguez
- Lisa Ellison, Sam Snead, and Maria Rodriguez

3)
```
SELECT Name
FROM Compensation
WHERE NOT(Salary > 30000 AND
BONUS > 1000);
```

- No names are selected
- Lisa Ellison
- Lisa Ellison, Sam Snead, and Maria Rodriguez

PARTICIPATION ACTIVITY 35.7.11: Select songs with NULL title or artist.

The given SQL creates a Song table and inserts some songs. The SELECT statement selects all songs.

Modify the SELECT statement to select only songs that have a NULL Title or NULL Artist.

Run your solution and verify only the songs with IDs 100, 300, 500, and 600 appear in the result table.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

```
1 CREATE TABLE Song (
2   ID INT,
3   Title VARCHAR(60),
4   Artist VARCHAR(60),
5   ReleaseYear INT
6 );
7
8 INSERT INTO Song VALUES
```

```
INSERT INTO Song VALUES
  (100, 'Hey Jude', NULL, 1968),
  (200, 'When Doves Cry', 'Prince', 1997),
  (300, NULL, 'The Righteous Brothers', 1964),
  (400, 'Johnny B. Goode', 'Chuck Berry', 1958),
  (500, 'Smells Like Teen Spirit', NULL, 1991),
  (600, NULL, 'Aretha Franklin', 1967);

-- Modify the SELECT statement
SELECT *
FROM Song;
```

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Run Reset code

▶ View solution

CHALLENGE ACTIVITY 35.7.1: Null values.

539740.3879454.qx3zqy7

Start

```
CREATE TABLE Country (
   15to64PopPct FLOAT,
   PopDensity DECIMAL(7, 2) NOT NULL,
   Over65PopPct FLOAT NOT NULL,
   ISOCode3 CHAR(3),
   ContinentCode CHAR(2),
   Under14PopPct FLOAT
);
```

Which columns can contain NULL values?

- ☐ 15to64PopPct
- ☐ PopDensity
- ☐ Over65PopPct
- ☐ ISOCode3
- ☐ ContinentCode
- ☐ Under14PopPct

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

1 2 3 4

Check Next

Exploring further:

- MySQL null values

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

35.8 Inserting, updating, and deleting rows

INSERT statement

The ***INSERT*** statement adds rows to a table. The INSERT statement has two clauses:

- The ***INSERT INTO*** clause names the table and columns where data is to be added. The keyword INTO is optional.
- The ***VALUES*** clause specifies the column values to be added.

The VALUES clause may list any number of rows in parentheses to insert multiple rows.

Figure 35.8.1: INSERT syntax.

```
INSERT [INTO] TableName (Column1, Column2,
...)
VALUES (Value1, Value2, ...);
```

PARTICIPATION ACTIVITY 35.8.1: Inserting rows into the Employee table.

table name column names

```
INSERT INTO Employee (ID, Name, Salary)
VALUES (2538, 'Lisa Ellison', 45000);
```

column values

```
INSERT INTO Employee
VALUES (5384, 'Sam Snead', 30500);
```

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Employee

ID	Name	Salary
2538	Lisa Ellison	45000
5384	Sam Snead	30500
6381	Maria Rodriguez	92300
7920	Jiho Chen	56000
9385	Alexis Parsons	32000

```
INSERT INTO Employee
VALUES (6381, 'Maria Rodriguez', 92300),
       (7920, 'Jiho Chen', 56000),
       (9385, 'Alexis Parsons', 32000);
```

Animation content:

Static figure:
The Employee table appears, with columns ID, Name, and Salary. Employee has five rows.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

A SQL statement appears:
Begin SQL code:
INSERT INTO Employee (ID, Name, Salary)
VALUES (2538, 'Lisa Ellison', 45000)
End SQL code.

A second SQL statement appears below the first:
Begin SQL code:
INSERT INTO Employee
VALUES (5384, 'Sam Snead', 30500)
End SQL code.

A third SQL statement appears below the second:
Begin SQL code:
INSERT INTO Employee
VALUES (6381, 'Maria Rodriguez', 92300),
 (7920, 'Jiho Chen', 56000),
 (9385, 'Alexis Parsons', 32000);
End SQL code.

Step 1: The INSERT statement's INTO clause names the Employee table and Employee's columns in parentheses. The INSERT INTO clause of the first statement appears. The label "table name" appears above Employee. The label "column names" appears above (ID, Name, Salary). The Employee table is empty.

Step 2: The VALUES clause names the column values in parenthesis. The value order must match the column order in the INTO clause. The VALUES clause of the first statement appears. The label "column values" appears above (2538, 'Lisa Ellison', 45000). One row is added to Employee:
2538, Lisa Ellison, 45000

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Step 3: The column names may be omitted as long as the VALUES clause lists all column values in the same order as the table's columns. The second statement appears. Another row is added to Employee:
5384, Sam Snead, 30500

Step 4: Any number of rows may be added with a single INSERT statement. The third statement appears. Three rows are added to Employee:
6381, Maria Rodriguez, 92300
7920, Jiho Chen, 56000
9385, Alexis Parsons, 32000

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Animation captions:

1. The INSERT statement's INTO clause names the Employee table and Employee's columns in parentheses.
2. The VALUES clause names the column values in parenthesis. The value order must match the column order in the INTO clause.
3. The column names may be omitted as long as the VALUES clause lists all column values in the same order as the table's columns.
4. Any number of rows may be added with a single INSERT statement.

PARTICIPATION ACTIVITY 35.8.2: INSERT statement.

Refer to the table definition below.

```
CREATE TABLE Department (
   Code TINYINT UNSIGNED,
   Name VARCHAR(20),
   ManagerID SMALLINT UNSIGNED
);
```

1) Which statement correctly inserts Engineering?

- ```
  INSERT INTO Department
  Code, Name, ManagerID
  VALUES (44, 'Engineering',
  2538);
  ```
- ```
  INSERT INTO (Code, Name,
  ManagerID)
  VALUES (44, 'Engineering',
  2538);
  ```
- ```
  INSERT INTO Department
  (Code, Name, ManagerID)
  VALUES (44, 'Engineering',
  2538);
  ```

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024
  ```

2) Which statement correctly inserts Sales?

-
```
INSERT Department
VALUES (82, 'Sales',
6381);
```

-
```
INSERT INTO Department
VALUES ('Sales', 82,
6381);
```

-
```
INSERT INTO Department
(82, 'Sales', 6381);
```

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

3) Which statement correctly inserts Marketing?

-
```
INSERT INTO Department
(Name, ManagerID, Code)
VALUES (12, 'Marketing',
6381);
```

-
```
INSERT INTO Department
(Name, ManagerID, Code)
VALUES ('Marketing', 6381,
12);
```

-
```
INSERT INTO Department
(Name, ManagerID, Code)
VALUES (6381, 12,
'Marketing');
```

4) Which statement correctly inserts Technical Support?

-
```
INSERT INTO Department
(Code, Name)
VALUES (99);
```

-
```
INSERT INTO Department
(Code, Name)
VALUES (99, 'Technical
Support');
```

-
```
INSERT INTO Department
(Code, Name)
VALUES (99, 'Technical
Support', NULL);
```

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

DEFAULT values

Columns may be omitted from an INSERT statement. When omitted, a column is assigned a NULL value. If the NOT NULL constraint is specified on the column, the insert is rejected.

Alternatively, a default value may be specified for a column. The optional ***DEFAULT*** keyword and default value follow the column name and data type in a CREATE TABLE statement. The column is assigned the default value, rather than NULL, when omitted from an INSERT statement.

PARTICIPATION ACTIVITY 35.8.3: DEFAULT values.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

```
CREATE TABLE Employee (
  ID          SMALLINT UNSIGNED,
  Name        VARCHAR(60),
  BirthDate   DATE DEFAULT '2000-01-01',
  Salary      DECIMAL(7,2) DEFAULT 0.00
);

INSERT INTO Employee (ID, BirthDate, Salary)
VALUES (6381, '1970-12-01', 980.00);

INSERT INTO Employee (ID, Name, Salary)
VALUES (2538, 'Lisa Ellison', 423.00);

INSERT INTO Employee (ID, Name, BirthDate)
VALUES (5384, 'Sam Snead', '2000-10-31');
```

Employee

ID	Name	BirthDate	Salary
6381	NULL	1970-12-01	980.0(
2538	Lisa Ellison	2000-01-01	423.0(
5384	Sam Snead	2000-10-31	0.00

Animation content:

Static figure:
Four SQL statements appear:
Begin SQL code:
CREATE TABLE Employee (
ID SMALLINT UNSIGNED,
Name VARCHAR(60),
BirthDate DATE DEFAULT '2000-01-01',
Salary DECIMAL(7,3) DEFAULT 0.00
);

INSERT INTO Employee (ID, BirthDate, Salary)
VALUES (6381, '1970-12-01', 980.00)

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

INSERT INTO Employee (ID, Name, Salary)
VALUES (2538, 'Lisa Ellison', 423.00)

INSERT INTO Employee (ID, Name, BirthDate)
VALUES (5384, 'Sam Snead', '2000-10-31')
End SQL code.

An Employee table appears, with columns ID, Name, BirthDate, and Salary. Employee has three rows:
6381, NULL, 1970-12-01, 980.00
2538, Lisa Ellison, 2000-01-01, 423.00
5384, Sam Snead, 2000-10-31, 0.00

Step 1: The Name column does not have a default value and is assigned NULL when omitted from an insert. The first two statements appear. In the CREATE TABLE statement, Name VARCHAR(60) is highlighted. One row is added to Employee:
6381, NULL, 1970-12-01, 980.00

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Step 2: The BirthDate column is assigned a default value of January 1, 2000 when omitted from an insert. The third statement appears. In the CREATE TABLE statement BirthDate clause, DEFAULT '2000-01-01' is highlighted. One row is added to Employee:
2538, Lisa Ellison, 2000-01-01, 423.00

Step 3: The Salary column is assigned a default value of 0.00 when omitted from an insert. The fourth statement appears. In the CREATE TABLE statement Salary clause, DEFAULT 0.00 is highlighted. Another row is added to Employee:
5384, Sam Snead, 2000-10-31, 0.00

Animation captions:

1. The Name column does not have a default value and is assigned NULL when omitted from an insert.
2. The BirthDate column is assigned a default value of January 1, 2000 when omitted from an insert.

PARTICIPATION ACTIVITY 35.8.4: DEFAULT values.

Refer to the statement below.

```
CREATE TABLE Department (
   Code      SMALLINT UNSIGNED DEFAULT 1000,
   Name      VARCHAR(20),
   ManagerID SMALLINT
);
```

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

1) Which alteration to the CREATE TABLE statement sets the default Name to Accounting?

- `Name VARCHAR(20) DEFAULT Accounting,`
- `Name VARCHAR(20) DEFAULT 'Accounting',`
- `Name VARCHAR(20) 'Accounting',`

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

2) What columns are NULL after the following statement executes?

```
INSERT INTO Department (Code)
VALUES (924);
```

- Name only
- ManagerID only
- Name and ManagerID

UPDATE statement

The ***UPDATE*** statement modifies existing rows in a table. The UPDATE statement uses the ***SET*** clause to specify the new column values. An optional WHERE clause specifies which rows are updated. Omitting the WHERE clause results in all rows being updated.

Figure 35.8.2: UPDATE syntax.

```
UPDATE TableName
SET Column1 = Value1, Column2 = Value2,
...
WHERE condition;
```

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

PARTICIPATION ACTIVITY 35.8.5: Updating rows in the Employee table.

```
UPDATE Employee
SET Name = 'Tom Snead',
```

Employee

ID	Name	BirthDate	Salary

```
    BirthDate = '2000-03-15'
WHERE ID = 5384;

UPDATE Employee
SET Salary = 42000;
```

2538	Lisa Ellison	1993-10-02	42000
5384	Tom Snead	2000-03-15	42000
6381	Maria Rodriguez	2001-12-21	42000

Animation content:

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Static figure:
Two SQL statements appear.
Begin SQL code:
UPDATE Employee
SET Name = 'Tom Snead', BirthDate = '2000-03-15'
WHERE ID = 5384;

UPDATE Employee
SET Salary = 42000;
End SQL code.

An Employee table appears, with columns ID, Name, BirthDate, and Salary. Employee has three rows.

Step 1: The UPDATE statement changes the Name and BirthDate of employee 5384. The first statement appears. The second row of Employee changes from:
5384, Sam Snead, 1995-03-15, 45000
to:
5384, Tom Snead, 2000-03-15, 45000

Step 2: The UPDATE statement has no WHERE clause, so the Salary in all rows is changed. The second statement appears. All values in the Salary column change to 42000.

Animation captions:

1. The UPDATE statement changes the Name and BirthDate of employee 5384.
2. The UPDATE statement has no WHERE clause, so the Salary in all rows is changed.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

PARTICIPATION ACTIVITY 35.8.6: Update the Song table.

The given SQL creates a Song table and inserts three songs.

Write three UPDATE statements to make the following changes:

- Change the title from 'One' to 'With Or Without You'.

- Change the artist from 'The Righteous Brothers' to 'Aretha Franklin'.
- Change the release years of all songs after 1990 to 2021.

Run your solution and verify the songs in the result table reflect the changes above.

```
CREATE TABLE Song (
  ID INT,
  Title VARCHAR(60),
  Artist VARCHAR(60),
  ReleaseYear INT
);

INSERT INTO Song VALUES
  (100, 'Blinding Lights', 'The Weeknd', 2019),
  (200, 'One', 'U2', 1991),
  (300, 'You\'ve Lost That Lovin\' Feeling', 'The Righteous Brothers', 
  (400, 'Johnny B. Goode', 'Chuck Berry', 1958);

-- Write your UPDATE statements here:



SELECT *
FROM Song;
```

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Run Reset code

▶ View solution

PARTICIPATION ACTIVITY 35.8.7: UPDATE statement.

Refer to the Department table.

Department

Code	Name	ManagerID
44	Engineering	2538
82	Sales	6381
12	Marketing	6381
99	Technical Support	NULL

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

1) What is missing to change "Sales" to "Custodial"?

```
UPDATE Department
SET ___
WHERE Code = 82;
```

- 'Sales' = 'Custodial'
- Department = 'Custodial'
- Name = 'Custodial'

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

2) What departments are changed?

```
UPDATE Department
SET Name = 'Administration'
WHERE ManagerID IS NOT NULL;
```

- Administration
- All departments except Technical Support
- All departments

3) What is missing to change Marketing's Code to 55 and Name to Administration?

```
UPDATE Department
SET Code = 55, Name =
'Administration'
WHERE ___;
```

- Code = 55
- ManagerID = 6381
- Code = 12

4) What department managers are changed?

```
UPDATE Department
SET ManagerID = 2538;
```

- All managers
- No managers
- Only the Engineering manager

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

DELETE statement

The ***DELETE*** statement deletes existing rows in a table. The ***FROM*** keyword is followed by the table name whose rows are to be deleted. An optional WHERE clause specifies which rows should be deleted. Omitting the WHERE clause results in all rows in the table being deleted.

Figure 35.8.3: DELETE syntax.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

```
DELETE FROM
TableName
WHERE condition;
```

PARTICIPATION ACTIVITY 35.8.8: Deleting rows from the Employee table.

```
DELETE FROM Employee
WHERE ID = 6381;
```

Employee

ID	Name	BirthDate	Salary
2538	Lisa Ellison	1993-10-02	45000
5384	Sam Snead	1995-03-15	30500
~~6381~~	~~Maria Rodriguez~~	~~2001-12-21~~	~~72300~~

```
DELETE FROM Employee
WHERE Salary > 40000 AND
      Salary < 80000;
```

Employee

ID	Name	BirthDate	Salary
~~2538~~	~~Lisa Ellison~~	~~1993-10-02~~	~~45000~~
5384	Sam Snead	1995-03-15	30500
~~6381~~	~~Maria Rodriguez~~	~~2001-12-21~~	~~72300~~

```
DELETE FROM Employee;
```

Employee

ID	Name	BirthDate	Salary
~~2538~~	~~Lisa Ellison~~	~~1993-10-02~~	~~45000~~
~~5384~~	~~Sam Snead~~	~~1995-03-15~~	~~30500~~
~~6381~~	~~Maria Rodriguez~~	~~2001-12-21~~	~~72300~~

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Animation content:

Static figure:
Three SQL statements appear.
Begin SQL code:
DELETE FROM Employee
WHERE ID equals 6381;

```
DELETE FROM Employee
WHERE Salary > 40000 AND Salary < 80000;

DELETE FROM Employee;
End SQL code.
```

Three copies of the Employee table appear to the right of the three SQL statements. Each copy has three rows and columns ID, Name, BirthDate, and Salary.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Step 1: The DELETE statement deletes the row with ID 6381 from the Employee table. The first statement appears, along with one copy of Employee. Row three of column ID has the value 6381. Row three is crossed out.

Step 2: The DELETE statement deletes rows where the Salary is between 40,000 and 80,000. Lisa and Maria are deleted. The second statement appears, along with another copy of Employee. Rows one and five of column Salary are highlighted and contain the values 45000 and 72400. Because these values satisfy the condition Salary > 40000 AND Salary < 80000 the, rows one and five are crossed out.

Step 3: The DELETE statement has no WHERE clause, so all employees are deleted. The third statement appears, along with the third copy of Employee. All rows are crossed out.

Animation captions:

1. The DELETE statement deletes the row with ID 6381 from the Employee table.
2. The DELETE statement deletes rows where the Salary is between 40,000 and 80,000. Lisa and Maria are deleted.
3. The DELETE statement has no WHERE clause, so all employees are deleted.

PARTICIPATION ACTIVITY 35.8.9: DELETE statement.

Refer to the Department table.

Department

Code	Name	ManagerID
44	Engineering	2538
82	Sales	6381
12	Marketing	6381
99	Technical Support	NULL

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

1) What departments are deleted?

```
DELETE FROM Department
WHERE ManagerID = 6381;
```

- Sales only
- Marketing only
- Sales and Marketing

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

2) What departments are deleted?

```
DELETE FROM Department;
```

- All departments
- No departments
- Engineering only

3) What is missing to delete only Sales?

```
DELETE FROM Department
WHERE ___;
```

- ManagerID = 6381
- Code = 82
- Code = 'Sales'

TRUNCATE

*The **TRUNCATE** statement deletes all rows from a table. TRUNCATE is nearly identical to a DELETE statement with no WHERE clause except for minor differences that depend on the database system.*

```
TRUNCATE TABLE TableName;
```

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

CHALLENGE ACTIVITY 35.8.1: Inserting, updating, and deleting rows.

539740.3879454.qx3zqy7

Start

Refer to the CREATE TABLE statement:

```
CREATE TABLE Country (
  Population INT,
  ISOCode2 CHAR(2),
  Name VARCHAR(50)
);
```

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Complete the statement to insert a row with Name 'Jordan', Population 9956011, and ISOCod

```
INSERT INTO Country (Population, ISOCode2, Name)
/* Your code here */ ;
```

1 2 3 4

Check Next

Exploring further:

- MySQL INSERT syntax
- MySQL UPDATE syntax
- MySQL DELETE syntax

35.9 Primary keys

Primary keys

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

A ***primary key*** is a column, or group of columns, used to identify a row. To ensure that each value identifies exactly one row, a primary key must be unique and not NULL.

In table diagrams, a bullet (●) precedes the primary key. The primary key is usually the left-most column of a table, but the position is not significant to the database. Ex: ID is the primary key of the Employee table below.

Table 35.9.1: Employee table with primary key.

Employee

● ID	Name	Salary
2538	Lisa Ellison	45000
5384	Sam Snead	30500
6381	Maria Rodriguez	92300

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Often, primary key values are used in the WHERE clause to select a specific row.

Figure 35.9.1: Primary key used to select a specific row.

```
SELECT Name
FROM Employee
WHERE ID = 5384;
```

```
Sam Snead
```

PARTICIPATION ACTIVITY 35.9.1: Primary keys.

Refer to the Employee table:

● ID	Name	Salary
2538	Lisa Ellison	45000
5384	Sam Snead	30500
6381	Maria Rodriguez	92300

1) Name is a good primary key.

- ○ True
- ○ False

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

2) A new employee can be added without an ID value.

- ○ True
- ○ False

3) A new employee can be added with ID 5384.

- ○ True
- ○ False

Composite primary keys

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Sometimes multiple columns are necessary to identify a row. A ***simple primary key*** consists of a single column. A ***composite primary key*** consists of multiple columns. A composite primary key must be:

- *Unique*. Values of primary key columns, when grouped together, must be unique. No group of values may repeat in multiple rows.
- *Not NULL*. No column of a composite primary key may contain a NULL value.
- ***Minimal***. All primary key columns are necessary for uniqueness. When any column is removed, the resulting simple or composite column is no longer unique.

Simple primary keys are necessarily minimal, since no column can be removed from a simple key.

In text, composite primary keys are enclosed in parentheses. Ex: (ColumnA, ColumnB). In table diagrams, a bullet (•) precedes every column of a composite primary key.

PARTICIPATION ACTIVITY 35.9.2: Composite primary keys.

Composite primary key

Family

• ID	• Number	Relationship	Name
2538	1	Spouse	Henry Ellison
2538	2	Son	Edward Ellison
6381	1	Spouse	Jose Rodriguez
6381	2	Daughter	Gina Rodriguez
6381	3	Daughter	Clara Rodriguez

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Animation content:

Step 1: ID is not unique in the Family table, since one employee may have several family members. A table Family appears with columns ID, Number, Relationship, and Name. Rows one and two of column ID are highlighted and both contain the value 2538. Rows three, four, and five of column ID are highlighted and all contain the value 6381.

Step 2: ID and Number together is unique, so (ID, Number) is a composite primary key. Row one of columns ID and Number are highlighted and contain values 2583 and 1 respectively. Row two of columns ID and Number is highlighted and contains the values 2583 and 2 respectively. Row three of columns ID and Number is highlighted and contains the values of 6381 and 1 respectively. Row four of columns ID and Number is highlighted and contains the values 6381 and 2 respectively. Row five of columns ID and Number is highlighted contains the values 6381 and 3 respectively.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Step 3: (ID, Number, Relationship) is unique. However, the Relationship column is unnecessary, so (ID, Number, Relationship) is not minimal. Row one of columns ID Number and Relationship is highlighted and contains the values 2583 1 and Spouse respectively. Row two of columns ID Number and Relationship is highlighted and contains the values 2583 2 and Son respectively. Row three of columns ID Number and Relationship is highlighted contains the values 6381 1 and Spouse respectively. Row four of columns ID Number and Relationship is highlighted and contains the values 6381 2 and Daughter respectively. Row five of columns ID Number and Relationship is highlighted and contains the values 6381 3 and Daughter respectively.

Animation captions:

1. ID is not unique in the Family table, since one employee may have several family members.
2. ID and Number together is unique, so (ID, Number) is a composite primary key.
3. (ID, Number, Relationship) is unique. However, the Relationship column is unnecessary, so (ID, Number, Relationship) is not minimal.

PARTICIPATION ACTIVITY 35.9.3: Composite primary keys.

Refer to the tables below.

Family

• ID	• Number	Relationship	Name
2538	1	Spouse	Henry Ellison
2538	2	Son	Edward Ellison
6381	1	Spouse	Jose Rodriguez
6381	2	Daughter	Gina Rodriguez
6381	3	Daughter	Clara Rodriguez

PhoneNumber

AreaCode	Exchange	Number
510	899	1111
212	899	1111
510	899	1234
212	611	1111

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

1) Can (ID, Relationship) be the primary key of Family?

- Yes
- No
- Cannot determine answer from data in the table.

2) The primary key of the PhoneNumber table is not indicated with a bullet. What is the primary key of PhoneNumber?

- Table has no primary key
- (AreaCode, Number)
- (AreaCode, Exchange, Number)

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

PRIMARY KEY constraint

The ***PRIMARY KEY*** constraint in a CREATE TABLE statement names the table's primary key. This constraint ensures that a column or group of columns is always unique and non-null.

In a CREATE TABLE statement, the primary key column definition usually appears first, followed by other column definitions and the primary key constraint. However, the order of CREATE TABLE clauses is not significant.

PARTICIPATION ACTIVITY 35.9.4: Adding primary key constraints to tables.

```
CREATE TABLE Employee (
   ID         SMALLINT UNSIGNED,
   Name       VARCHAR(60),
   Salary     DECIMAL(7,2),
   PRIMARY KEY (ID)
);
```

Employee

• ID	Name	Salary
2538	Lisa Ellison	45000
5384	Sam Snead	30400
6381	Maria Rodriguez	92300

```
CREATE TABLE Family (
   ID           SMALLINT UNSIGNED,
   Number       SMALLINT UNSIGNED,
   Relationship VARCHAR(20),
   Name         VARCHAR(60),
   PRIMARY KEY(ID, Number)
);
```

Family

• ID	• Number	Relationship	Name
2538	1	Spouse	Henry Ellison
2538	2	Son	Edward Ellison
6381	1	Spouse	Jose Rodriguez
6381	2	Daughter	Gina Rodriguez
6381	3	Daughter	Clara Rodriguez

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Animation content:

Static figure:
An SQL statement appears.
Begin SQL code:
CREATE TABLE Employee (

```
  ID SMALLINT UNSIGNED,
  Name VARCHAR(60),
  Salary DECIMAL(7,2),
  PRIMARY KEY (ID)
);
```
End SQL code.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

An Employee table appears to the right of this statement, with columns ID, Name, and Salary. The table has three rows.

A second SQL statement appears.
Begin SQL code:

```
CREATE TABLE Family (
  ID SMALLINT UNSIGNED,
  Number SMALLINT UNSIGNED,
  Relationship VARCHAR(20),
  Name VARCHAR(60),
  PRIMARY KEY(ID, Number)
);
```
End SQL code.

A Family table appears to the right of this statement, with columns ID, Number, Relationship, and Name. The table has five rows.

Step 1: The CREATE TABLE statement uses the keywords PRIMARY KEY to indicate the ID column is the table's primary key. The first SQL statement appears. The clause PRIMARY KEY (ID) is highlighted. The Employee table appears with a bullet next to the ID column. Employee has no rows.

Step 2: All rows added to the Employee table must have a unique ID. Three rows are added to Employee. All rows have different values in the ID column.

Step 3: The PRIMARY KEY constraint identifies the ID and Number columns as the Family table's composite primary key. The second SQL statement appears. The clause PRIMARY KEY (ID, Number) is highlighted. The Family table appears with bullets before columns ID and Number. Family has no rows.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Step 4: All rows added to the Family table must have a unique combination of ID and Number. Five rows are added to Family. All rows have different values in the composite column (ID, Number):
2538, 1
2538, 2
6381, 1
6381, 2
6381, 3

Animation captions:

1. The CREATE TABLE statement uses the keywords PRIMARY KEY to indicate the ID column is the table's primary key.
2. All rows added to the Employee table must have a unique ID.
3. The PRIMARY KEY constraint identifies the ID and Number columns as the Family table's composite primary key.
4. All rows added to the Family table must have unique combinations of ID and Number.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

PARTICIPATION ACTIVITY 35.9.5: PRIMARY KEY constraints.

Refer to the animation above.

1) Lisa, Sam, and Maria must have unique IDs and names.
   - True
   - False

2) The PRIMARY KEY constraint may include multiple columns.
   - True
   - False

3) Assuming the Family table has the five rows shown above, a new row with values (2538, 2, 'Daughter', 'Ella Ellison') may be added to the Family table.
   - True
   - False

Auto-increment columns

An ***auto-increment column*** is a numeric column that is assigned an automatically incrementing value when a new row is inserted. The ***AUTO_INCREMENT*** keyword defines an auto-increment column. AUTO_INCREMENT follows the column's data type in a CREATE TABLE statement.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Integer primary keys are commonly implemented as auto-increment columns. In MySQL, AUTO_INCREMENT may be applied only to primary key columns.

Figure 35.9.2: Auto-increment primary key example.

```
CREATE TABLE Employee (
   ID        SMALLINT UNSIGNED
AUTO_INCREMENT,
   Name      VARCHAR(60),
   BirthDate DATE,
   Salary    DECIMAL(7,2),
   PRIMARY KEY (ID)
);
```

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Database users occasionally make the following errors when inserting primary keys:

- Inserting values for auto-increment primary keys.
- Omitting values for primary keys that are not auto-increment columns.

MySQL allows insertion of a specific value to an auto-increment column. However, overriding auto-increment for a primary key is usually a mistake.

PARTICIPATION ACTIVITY 35.9.6: Common INSERT errors.

✗
```
INSERT INTO Employee
VALUES (2538, 'Maria Rodriguez', 92300);
```

✓
```
INSERT INTO Employee
VALUES (6381, 'Maria Rodriguez', 92300);
```

Employee

ID	Name	Salary
2538	Lisa Ellison	45000
5384	Sam Snead	30500
6381	Maria Rodriguez	92300

✗
```
INSERT INTO Employee
VALUES (3, 'Maria Rodriguez', 92300);
```

✓
```
INSERT INTO Employee (Name, Salary)
VALUES ('Maria Rodriguez', 92300);
```

auto-increment

Employee

ID	Name	Salary
1	Lisa Ellison	45000
2	Sam Snead	30500
3	Maria Rodriguez	92300

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

✗
```
INSERT INTO Employee (Name, Salary)
VALUES ('Maria Rodriguez', 92300);
```

✓
```
INSERT INTO Employee (ID, Name, Salary)
VALUES (6381, 'Maria Rodriguez', 92300);
```

non auto-increment

Employee

ID	Name	Salary
2538	Lisa Ellison	45000
5384	Sam Snead	30500
6381	Maria Rodriguez	92300

Animation content:

Step 1: The INSERT statement uses an ID that already exists in Employee. Duplicate primary key values cannot be added, so a unique ID must be chosen. A table Employee appears, with columns ID, Name, and Salary. ID is a primary key. There are two lines of code. The first line of code states INSERT INTO Employee. Line two of code states VALUES left parenthesis 2538 comma apostrophe Maria Rodriguez apostrophe comma 92300 right parenthesis semicolon. 2538 in the second line of code is boxed and value 2538 in column ID of table Employee is highlighted. 2538 in the second line of code is crossed out and a red X is put next to these two lines of code. Two new lines of code appear. The first line of code states left parenthesis 6381 comma apostrophe Maria Rodriguez apostrophe comma 92300 right parenthesis semicolon. 6381 in the second line of code is boxed and values 6381 Maria Rodriguex and 92300 are added as a new row to columns ID Name and Salary respectively. A green checkmark is placed next to the code.

Step 2: If ID is an auto-increment column, the ID should not be listed in the INSERT statement. The database assigns the ID automatically. A second table Employee appears, with columns ID, Name, and Salary. ID is a primary key and is also labeled auto-increment. Two new lines of code appear. The first line one code states INSERT INTO Employee. The second line of code states VALUES left parenthesis 3 comma apostrophe Maria Rodriguex apostrophe comma 92300 right parenthesis semicolon. The 3 in the second line of code is boxed and then crossed out. A big red X appears next to these two lines of code. Two new lines of code appear. The first line of code states INSERT INTO Employee left parenthesis Name comma Salary right parenthesis. The second line of code states VALUES left parenthesis apostrophe Maria Rodriguex apostrophe comma 92300 right parenthesis semicolon. The values Maria Rodriguez and 92300 are added into a new row in columns Name and Salary of table Employee respectively. 3 is filled into the same row in column ID and is highlighted. A green check mark appears next to the two lines of code.

Step 3: If ID is not an auto-increment column, then an ID value must be specified. A third Employee table appears, with columns ID, Name, and Salary. ID is a primary key and is also labeled non auto-increment, and Salary is labeled NOT NULL. Two new lines of code appear. The first line one code states INSERT INTO Employee left parenthesis Name comma Salary right parenthesis. The second line of code states VALUES left parenthesis apostrophe Maria Rodriguez apostrophe comma 92300 right parenthesis semicolon. A big red X appears next to these two lines of code. Two new lines of code appear. The first line of code states INSERT INTO Employee left parenthesis ID comma Name comma Salary right parenthesis. The second line of code states VALUES left parenthesis 6381 comma apostrophe Maria Rodriguez apostrophe comma 92300 right parenthesis semicolon. The values 6381 Maria Rodriguez and 92300 are added into a new row in columns ID Name and Salary of table Employee respectively. A green check mark appears next to the two lines of code.

Animation captions:

1. The INSERT statement uses an ID that already exists in Employee. Duplicate primary key values cannot be added, so a unique ID must be chosen.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

2. If ID is an auto-increment column, the ID should not be listed in the INSERT statement. The database assigns the ID automatically.
3. If ID is not an auto-increment column, then an ID value must be specified.

PARTICIPATION ACTIVITY 35.9.7: Insert rows into Movie table.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

The given SQL creates a Movie table with an auto-incrementing ID column.

Write a single INSERT statement immediately after the CREATE TABLE statement that inserts the following movies:

Title	Rating	Release Date
Raiders of the Lost Ark	PG	June 15, 1981
The Godfather	R	March 24, 1972
The Pursuit of Happyness	PG-13	December 15, 2006

Note that dates above need to be converted into 'YYYY-MM-DD' format (with single quotes) in the INSERT statement.

Run your solution and verify the movies in the result table have the auto-assigned IDs 1, 2, and 3.

```
1
2  CREATE TABLE Movie (
3    ID INT AUTO_INCREMENT,
4    Title VARCHAR(100),
5    Rating CHAR(5) CHECK (Rating IN ('G', 'PG', 'PG-13', 'R')),
6    ReleaseDate DATE,
7    PRIMARY KEY (ID)
8  );
9
10 -- Write your INSERT statement here:
11
12
13
14 SELECT *
15 FROM Movie;
16
```

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Run Reset code

▶ View solution

PARTICIPATION ACTIVITY 35.9.8: Common INSERT errors.

Refer to the table definition below.

```
CREATE TABLE Department (
  Code TINYINT UNSIGNED AUTO_INCREMENT,
  Name VARCHAR(20) NOT NULL,
  ManagerID SMALLINT UNSIGNED,
  PRIMARY KEY (Code)
);
```

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

1) Which statement correctly inserts Engineering?

```
INSERT INTO Department
(Code, Name, ManagerID)
VALUES (44, 'Engineering',
2538);
```

```
INSERT INTO Department
VALUES ('Engineering',
2538);
```

```
INSERT INTO Department
(Name, ManagerID)
VALUES ('Engineering',
2538);
```

2) Which statement correctly inserts an unnamed department with no manager?

```
INSERT INTO Department
(Name, ManagerID)
VALUES ('', NULL);
```

```
INSERT INTO Department
VALUES (NULL, '', NULL);
```

```
INSERT INTO Department
(Name, ManagerID)
VALUES ('');
```

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

CHALLENGE ACTIVITY 35.9.1: Primary keys.

539740.3879454.qx3zqy7

Start

Country

● Code	CountryName	Continent	ContinentCode
430	Liberia	Africa	AF
56	Belgium	Europe	EU
634	Qatar	Asia	AS
50	Bangladesh	Asia	AS
548	Vanuatu	Oceania	OC

```
SELECT CountryName
FROM Country
WHERE Code = 56;
```

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

What is returned?

- [ ] Liberia
- [ ] Belgium
- [ ] Qatar
- [ ] Bangladesh
- [ ] Vanuatu

1 2 3 4 5

Check Next

35.10 Foreign keys

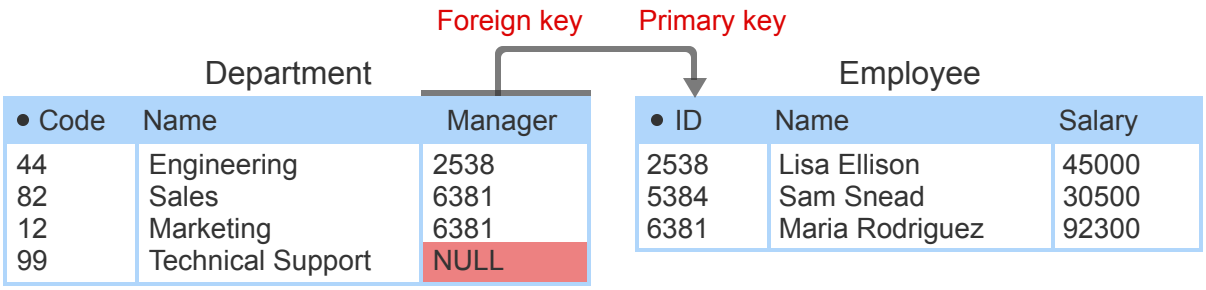
Foreign keys

A ***foreign key*** is a column, or group of columns, that refer to a primary key. The data types of the foreign and primary keys must be the same, but the names may be different. Unlike primary keys, foreign key values may be NULL and are not necessarily unique. However, a foreign key value that is not NULL must match some value of the referenced primary key.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio

In table diagrams, an arrow indicates a foreign key. The arrow starts at the foreign key and points to the table containing the referenced primary key.

PARTICIPATION ACTIVITY 35.10.1: The foreign key Manager refers to the primary key ID.



©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Animation content:

Step 1: The Department table's column is a foreign key that refers to the primary key ID in the Employee table. The Department table is on the left and the Employee table is on the right. Department has columns Code, Name, and Manager. Code is preceded by a solid circle. Department has four rows with values (44, Engineering, 2538), (82, Sales, 6381), (12, Marketing, 6381), and (99, Technical Support, NULL). Employee has columns ID, Name, and Salary. ID is preceded by a solid circle. Employee has three rows, with values (2538, Lisa Ellison, 45000), (5384, Sam Snead, 30500), and (6381, Maria Rodriguez, 92300). Manager is labeled Foreign key, ID is labeled Primary key, and an arrow points from Manager to ID.

Step 2: Lisa Ellison manages the engineering department. The value 2538 in row one of Department Manager, and in row one of Employee ID, is highlighted.

Step 3: Maria Rodriguez manages the sales and marketing departments. The value 6381 in rows two and three of Department Manager, and row three of Employee ID, is highlighted.

Step 4: The Technical Support department has no manager. The value NULL in row four Department Manager is highlighted.

Animation captions:

1. The Department table's Manager column is a foreign key that refers to the primary key ID in the Employee table.
2. Lisa Ellison manages the engineering department.
3. Maria Rodriguez manages the sales and marketing departments.
4. The Technical Support department has no manager.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

PARTICIPATION ACTIVITY 35.10.2: Foreign keys.

Refer to the tables above.

1) The data type of Manager and ID must be the same.
   - True
   - False

2) NULL in the Manager column refers to an Employee row with a NULL ID.
   - True
   - False

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

3) Sam Snead does not manage a department.
   - True
   - False

4) Replacing NULL in the Manager column with 5384 assigns Sam Snead as the manager of the Technical Support department.
   - True
   - False

5) The NULL in the Manager column may be replaced with 9876.
   - True
   - False

6) Values in a foreign key must be unique.
   - True
   - False

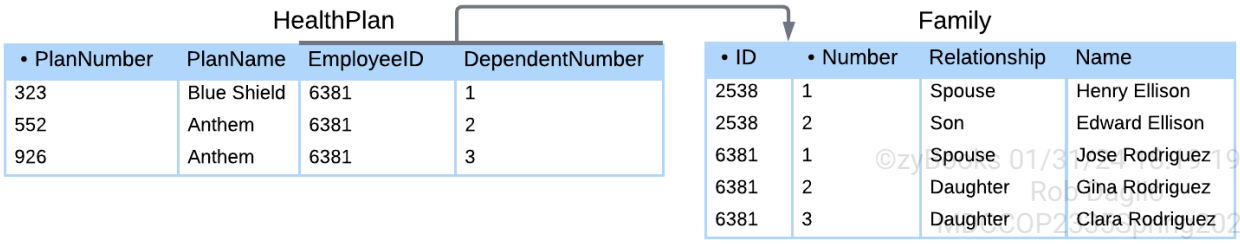
Composite foreign keys

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

A foreign key that refers to a composite primary key must also be composite. All columns of a composite foreign key value must either be NULL or match some primary key value. In table diagrams, the tail of the arrow extends across all columns of a composite foreign key.

In the figure below, the composite foreign key (EmployeeID, DependentNumber) of HealthPlan refers to the composite primary key ((ID, Number) of Family.

Figure 35.10.1: Composite foreign keys.



• PlanNumber	PlanName	EmployeeID	DependentNumber
323	Blue Shield	6381	1
552	Anthem	6381	2
926	Anthem	6381	3

• ID	• Number	Relationship	Name
2538	1	Spouse	Henry Ellison
2538	2	Son	Edward Ellison
6381	1	Spouse	Jose Rodriguez
6381	2	Daughter	Gina Rodriguez
6381	3	Daughter	Clara Rodriguez

©zyBooks 01/31/24 18:19 1939727 Rob Daglio MDCCOP2335Spring2024

PARTICIPATION ACTIVITY 35.10.3: Composite foreign keys.

Refer to the tables above.

1) Which family member has the Blue Shield health plan?

- Harry Ellison
- Jose Rodriguez
- Clara Rodriguez

2) Can the HealthPlan table contain the value (2538, NULL) in (EmployeeID, DependentNumber)?

- Yes, with the Family data shown above
- Yes, if Family had additional rows and primary key values
- No

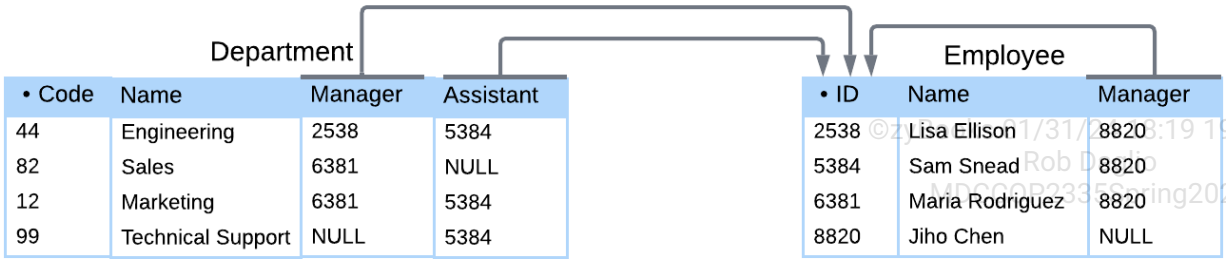
Special cases

©zyBooks 01/31/24 18:19 1939727 Rob Daglio MDCCOP2335Spring2024

Multiple foreign keys may refer to the same primary key. A foreign key may refer to the primary key of the same table.

In the figure below, the Manager and Assistant foreign keys of Department both refer to ID, the primary key of Employee. The Manager foreign key of Employee also refers to the primary key of Employee.

Figure 35.10.2: Special cases.



• Code	Name	Manager	Assistant
44	Engineering	2538	5384
82	Sales	6381	NULL
12	Marketing	6381	5384
99	Technical Support	NULL	5384

• ID	Name	Manager
2538	Lisa Ellison	8820
5384	Sam Snead	8820
6381	Maria Rodriguez	8820
8820	Jiho Chen	NULL

©zyBooks 01/31/24 18:19 1939727 Rob Daglio MDCCOP2335Spring2024

PARTICIPATION ACTIVITY 35.10.4: Special cases.

Refer to the tables above.

1) Which department has the same manager and assistant?

- Engineering
- Sales
- No department has the same manager and assistant.

2) Who is Lisa Ellison's manager?

- Sam Snead
- Maria Rodriguez
- Jiho Chen

Foreign key constraint

A foreign key constraint is created with a foreign key clause in the CREATE TABLE statement. The clause consists of the ***FOREIGN KEY*** keyword followed by the foreign key column, and the ***REFERENCES*** keyword followed by the referenced table and primary key column. The clause may appear anywhere in the CREATE TABLE statement, but usually follows all column definitions.

When a foreign key constraint is specified, the database rejects insert, update, and delete statements that violate referential integrity.

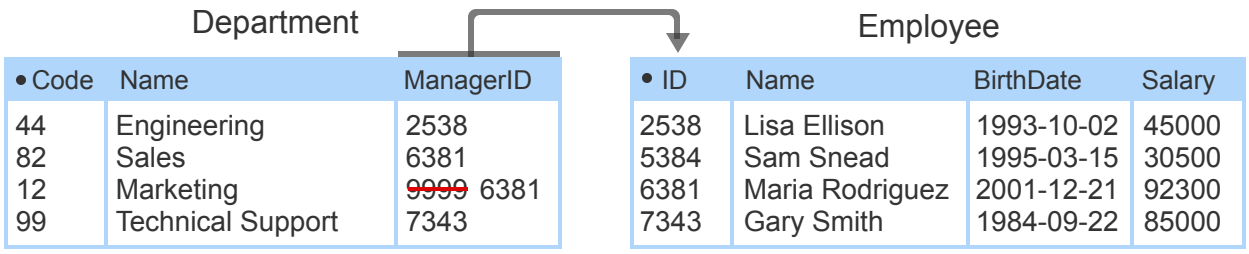
©zyBooks 01/31/24 18:19 1939727 Rob Daglio MDCCOP2335Spring2024

Figure 35.10.3: FOREIGN KEY syntax.

```
CREATE TABLE TableName (
  . . .
  FOREIGN KEY (ColumnName) REFERENCES TableName
(ColumnName),
  . . .
);
```

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

PARTICIPATION ACTIVITY 35.10.5: Foreign key constraint.




```
CREATE TABLE Department (
   Code      TINYINT UNSIGNED,
   Name      VARCHAR(20),
   ManagerID SMALLINT UNSIGNED,
   PRIMARY KEY (Code),
   FOREIGN KEY (ManagerID) REFERENCES Employee(ID)
);
```

Animation content:

Static figure:
Two tables appear. The Department table has columns Code, Name, and ManagerID. Code has a bullet. Department has four rows. The Employee table has columns ID, Name, BirthDate, and Salary. ID has a bullet. Employee has four rows. An arrow points from the ManagerID column to the ID column.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

An SQL statement appears.
Begin SQL code:
CREATE TABLE Department (
  Code TINYINT UNSIGNED,
  Name VARCHAR(20),

```
  ManagerID SMALLINT UNSIGNED,
  PRIMARY KEY (Code),
  FOREIGN KEY (ManagerID) REFERENCES Employee(ID)
);
```

End SQL code.

Step 1: The Employee table has primary key ID. The Employee table appears.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Step 2: The Department table is created with a FOREIGN KEY constraint that REFERENCES the Employee ID column. The FOREIGN KEY clause of the SQL statement is highlighted.The Department table appears with no rows. The arrow from ManagerID to ID appears.

Step 3: When rows are added to Department, the ManagerID value must exist in the ID column. 9999 does not appear in ID and is rejected. Four rows are added to the Department table:
Row one of column ManagerID has value 2538 and matches the value 2538 in the column ID of table Employee.
Row two of column ManagerID has value 6381 and matches the value 6381 in row three of column ID of table Employee.
Row three of column ManagerID has value 9999 and does not match any value in column ID of table Employee. Value 9999 is crossed out and changed to 6381, which matches the value in row three of column ID of table Employee.
Row four of column ManagerID has value 7343 and matches the value 7343 of column ID of table Employee.

Animation captions:

1. The Employee table has primary key ID.
2. The Department table is created with a FOREIGN KEY constraint that REFERENCES the Employee ID column.
3. When rows are added to Department, the ManagerID value must exist in the ID column. 9999 does not appear in ID and is rejected.

PARTICIPATION ACTIVITY 35.10.6: Add primary and foreign key constraints.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

The given SQL creates an Album table and a Song table. The CREATE TABLE statements do not specify primary or foreign key constraints. The SHOW COLUMNS queries show information about the Album and Song table columns.

Add three constraint clauses that specify primary and foreign keys, as follows:

1. Add a primary key constraint to the Album table's ID column.
2. Add a primary key constraint to the Song table's ID column.

3. Add a foreign key constraint to the Song table. This constraint specifies that the AlbumID column refers to the Album table's ID column.

After the constraint clauses are added, the SHOW COLUMNS result tables will show "PRI" in the Key column to indicate which Album and Song columns are primary keys, and "MUL" to indicate which Song column is the foreign key.

```
1  -- Add a primary key
2  CREATE TABLE Album (
3    ID INT,
4    Title VARCHAR(60),
5    ReleaseYear INT
6  );
7
8  -- Add primary and foreign keys
9  CREATE TABLE Song (
10   ID INT,
11   Title VARCHAR(60),
12   Artist VARCHAR(60),
13   AlbumID INT
14 );
15
16 SHOW COLUMNS
17 FROM Album;
18
19 SHOW COLUMNS
20 FROM Song;
```

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Run Reset code

▶ View solution

PARTICIPATION ACTIVITY 35.10.7: Foreign key constraint.

1) In a CREATE TABLE statement, the FOREIGN KEY constraint must follow all column declarations.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

- ○ True
- ○ False

2) In a FOREIGN KEY constraint, parentheses are required around the foreign key column name.

- True
- False

3) In a FOREIGN KEY constraint, data types of the foreign key and primary key columns must be the same.

- True
- False

4) Adding a FOREIGN KEY constraint to a table only affects inserting new rows into the table.

- True
- False

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

CHALLENGE ACTIVITY 35.10.1: Foreign key constraints.

539740.3879454.qx3zqy7

Start

Country

● ISOCode3	CountryName	PopDensity	Population
TKM	Turkmenistan	32.24689242	5850908
TLS	East Timor	220.8495228	1267972
URY	Uruguay	51.0435573	3449299

Geography

● ISOCode2	Code	IndependenceYear	Capital
TM	TKM	1991	Ashgabat
TL	TLS	2002	Dili
UY	URY	1825	Montevideo

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

What is the Capital of Uruguay?

1 2 3

Check Next

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Exploring further:

- MySQL FOREIGN KEY constraint

35.11 Referential integrity

Referential integrity rule

A ***fully NULL*** foreign key is a simple or composite foreign key in which all columns are NULL. ***Referential integrity*** is a relational rule that requires foreign key values are either fully NULL or match some primary key value.

In a relational database, foreign keys must obey referential integrity at all times. Occasionally, data entry errors or incomplete data result in referential integrity violations. Violations must be corrected before data is stored in the database.

PARTICIPATION ACTIVITY 35.11.1: Referential integrity violations.

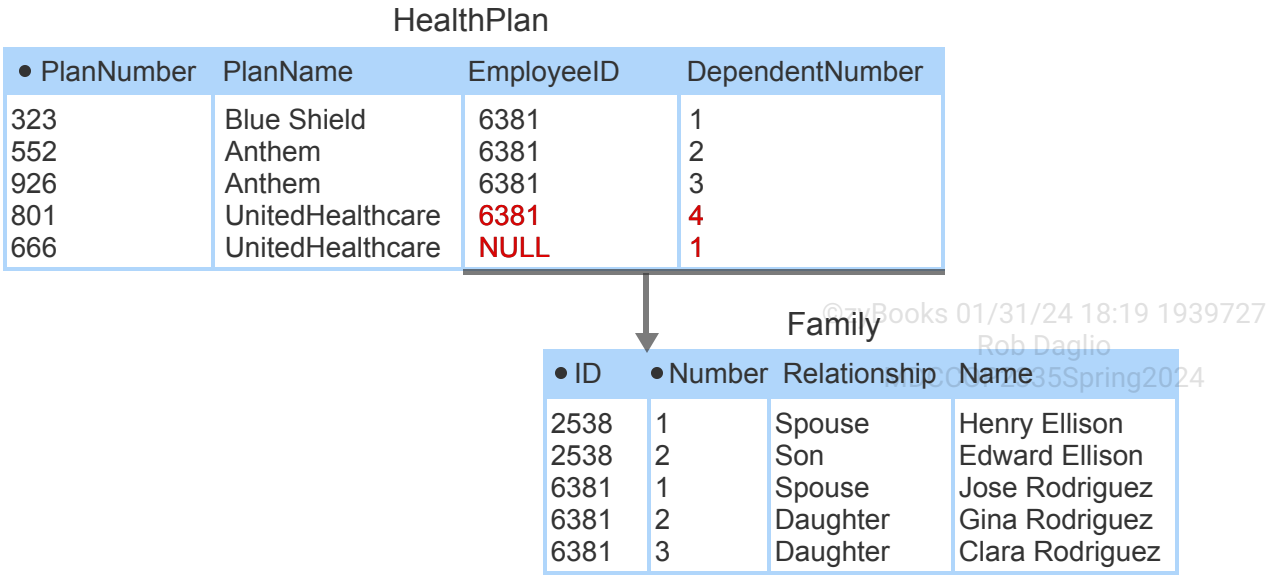
©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Department

• Code	Name	Manager
44	Engineering	2538
82	Sales	6381
12	Marketing	6381
99	Technical Support	NULL
23	Human resources	4407

Employee

• ID	Name	Salary
2538	Lisa Ellison	45000
5384	Sam Snead	30500
6381	Maria Rodriguez	92300



©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Animation content:

Step 1: 4407 does not match any value in ID and violates referential integrity. The Department and Employee tables appear. Department has columns Code, Name, and Manager. Employee has columns ID, Name, and Salary. ID is the primary key. An arrow points from the Manager column to the ID column. Row five of column Manager is highlighted and contains the value 4407.

Step 2: (6381, 4) does not match any value in (ID, Number) and violates referential integrity. The HealthPlan and Family tables appear. Health Plan has columns PlanNumber, PlanName, EmployeeID, and DependentNumber. Family has columns ID, Number, Relationship, and Name. (ID, Number) is the primary key. An arrow points from (EmployeeID, DependentNumber) of the HealthPlan table to (ID, Number) of the Family table. The value (6381, 4) in (EmployeeID, DependentNumber) is highlighted.

Step 3: (NULL, 1) is partially NULL and violates referential integrity. The value (NULL, 1) in (EmployeeID, DependentNumber) is highlighted.

Animation captions:

1. 4407 does not match any value in ID and violates referential integrity.
2. (6381, 4) does not match any value in (ID, Number) and violates referential integrity.
3. (NULL, 1) is partially NULL and violates referential integrity.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

PARTICIPATION ACTIVITY 35.11.2: Referential integrity rules for simple primary keys.

Refer to the tables below.

Department

• Code	Name	Manager
44	Engineering	2538
82	Sales	3829
12	Marketing	6381
99	Technical Support	NULL

Employee

• ID	Name	Salary
2538	Lisa Ellison	45000
5384	Sam Snead	30500
6381	Maria Rodriguez	92300

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

1) In the Department table, which foreign key value violates referential integrity?

- 2538
- 3829
- NULL

2) Does the NULL in the Manager column violate referential integrity?

- Yes
- No

PARTICIPATION ACTIVITY 35.11.3: Referential integrity rules for composite foreign keys.

Refer to the tables below.

HealthPlan

• PlanNumber	PlanName	EmployeeID	DependentNumber
323	Blue Shield	6381	1
552	Anthem	6381	2
926	Anthem	6381	3
801	UnitedHealthCare	6381	4
666	UnitedHealthCare	6381	NULL
744	Blue Shield	NULL	NULL

Family

• ID	• Number	Relationship	Name
2538	1	Spouse	Henry Ellison
2538	2	Son	Edward Ellison
6381	1	Spouse	Jose Rodriguez
6381	2	Daughter	Gina Rodriguez
6381	3	Daughter	Clara Rodriguez

1) In the HealthPlan table, which foreign key value violates referential integrity?

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

- (NULL, NULL) only
- (6381, NULL) only
- (6381, 4) only
- Both (6381, NULL) and (6381, 4)

2) In the HealthPlan table, which foreign key value is fully NULL?

- (6381, NULL)
- (NULL, NULL)
- No foreign key values are fully NULL

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Referential integrity violations

Referential integrity can be violated in four ways:

1. A primary key is updated.
2. A foreign key is updated.
3. A row containing a primary key is deleted.
4. A row containing a foreign key is inserted.

Only these four operations can violate referential integrity. Primary key inserts and foreign key deletes never violate referential integrity.

PARTICIPATION ACTIVITY 35.11.4: Four ways to violate referential integrity.

Department

• Code	Name	Manager
44	Engineering	2538
82	Sales	6381
12	Marketing	6381
99	Technical Support	NULL
49	Administration	0202

Employee

• ID	Name	Salary
2538	Lisa Ellison	45000
5384	Sam Snead	30500
6381	Maria Rodriguez	92300

Animation content:

Step 1: Updating the Employee primary key to 8888 violates referential integrity because the foreign key 2538 no longer exists in Employee. There are two tables named Department and Employee. Department has columns Code, Name, and Manager. Employee has columns ID, Name, and Salary. ID is the primary key. An arrow points from Manager to ID. Row one of columns Manager and ID are highlighted and both contain the value 2538. The value in row one of column ID changes from 2538 to 8888.

Step 2: Updating the foreign key to 3333 violates referential integrity because 3333 does not match a primary key value. Row two of column Manager and row three of column ID contain the value

6381 and are highlighted. The value in row two of column Manager changes from 6381 to 3333.

Step 3: Deleting Employee primary key 6381 violates referential integrity because the foreign key 6381 no longer exists in Employee. Row three of the Employee table is highlighted:
6381, Maria Rodriguez, and 92300
A line strikes through this row. Rows two and three of column Manager are highlighted and both contain the values 6381.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Step 4: Inserting foreign key 0202 violates referential integrity because 0202 does not match a primary key value.A new row is added to the Department table:
49, Administration, 0202
The value 0202 in this row is highlighted.

Animation captions:

1. Updating the Employee primary key to 8888 violates referential integrity because the foreign key 2538 no longer exists in Employee.
2. Updating the foreign key to 3333 violates referential integrity because 3333 does not match a primary key value.
3. Deleting Employee primary key 6381 violates referential integrity because the foreign key 6381 no longer exists in Employee.
4. Inserting foreign key 0202 violates referential integrity because 0202 does not match a primary key value.

PARTICIPATION ACTIVITY 35.11.5: Referential integrity violations.

Refer to the tables in the above animation. Match the violation type to the database change.

If unable to drag and drop, refresh the page.

Insert a foreign key **Delete a primary key** **Update a primary key**

Update a foreign key

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

	Change Maria Rodriguez's ID to 9925.
	Change the Technical Support department manager to 8001.

	Remove Lisa Ellison from the Employee table.
	Add Human Resources to the Department table with manager 1420.

Reset

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

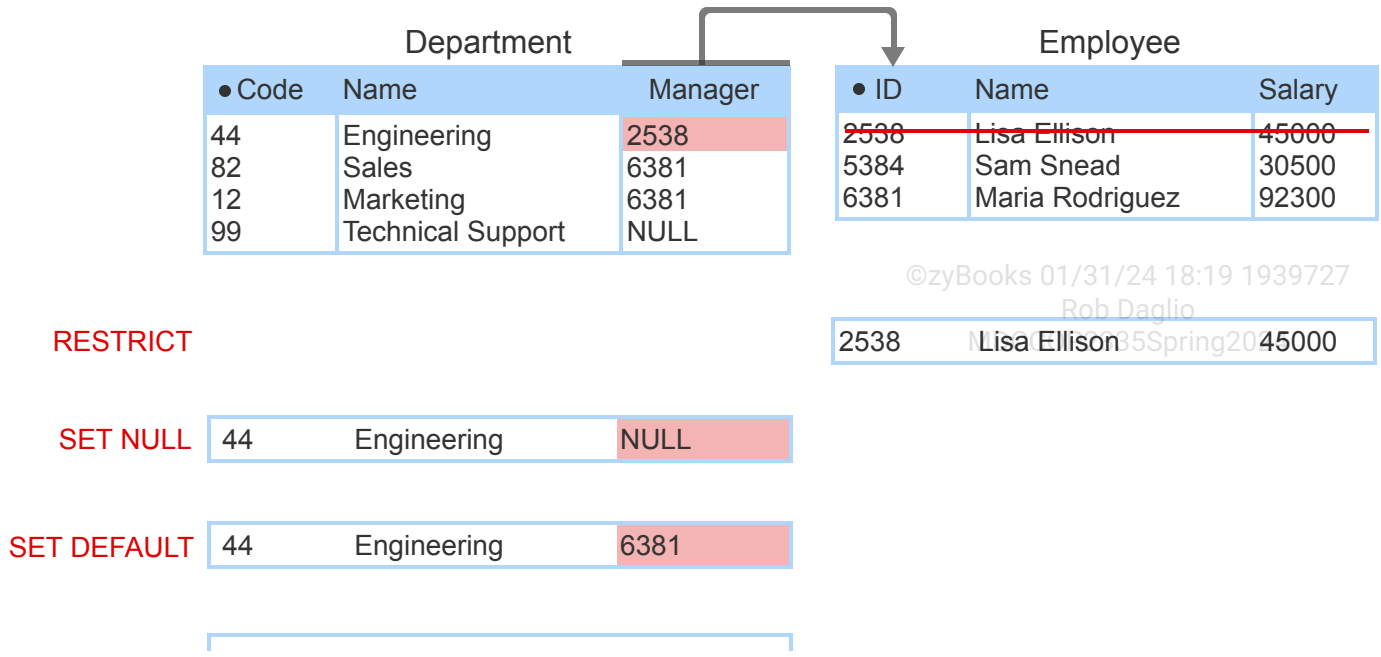
Referential integrity actions

An insert, update, or delete that violates referential integrity can be corrected manually. However, manual corrections are time-consuming and error-prone. Instead, databases automatically correct referential integrity violations with any of four actions, specified as SQL constraints:

- ***RESTRICT*** rejects an insert, update, or delete that violates referential integrity.
- ***SET NULL*** sets invalid foreign keys to NULL.
- ***SET DEFAULT*** sets invalid foreign keys to the foreign key default value.
- ***CASCADE*** propagates primary key changes to foreign keys.

CASCADE behaves differently for primary key updates and deletes. If a primary key is deleted, rows containing matching foreign keys are deleted. If a primary key is updated, matching foreign keys are updated to the same value.

PARTICIPATION ACTIVITY 35.11.6: Referential integrity actions on primary key delete.



©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

CASCADE	44	Engineering	2538

Animation content:

Static figure:
The Department table has columns Code, Name, and Manager. Code is the primary key. Department has four rows. The first row is:
44, Engineering, 2538

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

The Employee table has columns ID, Name, and Salary. ID is the primary key. Employee has three rows. The first row is:
2538, Lisa Ellison, 45000

An arrow points from the Manager column of Department to the ID column of Employee.

Step 1: The row containing primary key 2538 is deleted. A line strikes through the Employee row:
44, Engineering, 2538

Step 2: RESTRICT rejects the delete, since employee 2538 manages Engineering. A row labeled RESTRICT appears below the Employee table:
2538, Lisa Ellison, 45000
A line strikes through this row.

Step 3: SET NULL sets matching foreign keys to NULL. A row labeled SET NULL appears below the Department table:
44, Engineering, NULL
The NULL value is highlighted.

Step 4: SET DEFAULT sets matching foreign keys to the foreign key default value, 6381. A row labeled SET DEFAULT appears below the Department table:
44, Engineering, 6381
The value 6381 is highlighted.

Step 5: CASCADE deletes all rows with matching foreign key values. A line strikes through the Engineering row in the Department table. A row labeled CASCADE appears below the Department table:
44, Engineering, 2538
A line strikes through this row.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Animation captions:

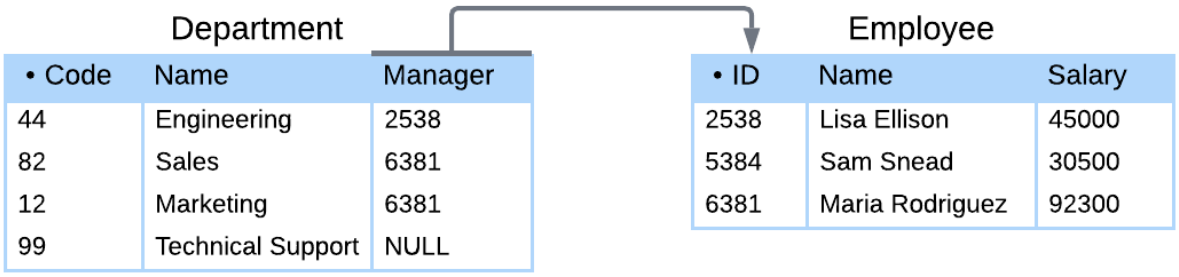
1. The row containing primary key 2538 is deleted.

2. RESTRICT rejects the delete, since employee 2538 manages Engineering.
3. SET NULL sets matching foreign keys to NULL.
4. SET DEFAULT sets matching foreign keys to the foreign key default value, 6381.
5. CASCADE deletes all rows with matching foreign key values.

PARTICIPATION ACTIVITY 35.11.7: Referential integrity actions.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Refer to the tables below. What are the results of the following actions?



Department

• Code	Name	Manager
44	Engineering	2538
82	Sales	6381
12	Marketing	6381
99	Technical Support	NULL

Employee

• ID	Name	Salary
2538	Lisa Ellison	45000
5384	Sam Snead	30500
6381	Maria Rodriguez	92300

1) RESTRICT, when the row containing Maria Rodriguez is deleted.

- ○ The Sales and Marketing managers are set to NULL.
- ○ The Sales and Marketing departments are deleted.
- ○ The delete is rejected.

2) SET NULL, when Lisa Ellison's ID is changed to 1001.

- ○ The Engineering manager is set to NULL.
- ○ The Engineering manager is set to 1001.
- ○ The change is rejected.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

3) SET DEFAULT, when Lisa Ellison's ID is changed to 1001.

- ○ The Engineering manager is set to NULL.
- ○ The Engineering manager is set to the Manager default value.
- ○ The change is rejected.

4) CASCADE, when Maria Rodriguez' ID is changed to 2022.

- The Sales and Marketing managers are set to NULL.
- The Sales and Marketing managers are set to 2022.
- The change is rejected.

5) CASCADE, when Maria Rodriguez is deleted.

- The Sales and Marketing managers are set to NULL.
- The Sales and Marketing departments are deleted.
- The delete is rejected.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

ON UPDATE and ON DELETE clauses

For foreign key inserts and updates, MySQL supports only RESTRICT. Foreign key inserts and updates that violate referential integrity are automatically rejected.

For primary key updates and deletes, MySQL supports all four actions. Actions are specified in the optional ***ON UPDATE*** and ***ON DELETE*** clauses of the FOREIGN KEY constraint. ON UPDATE and ON DELETE are followed by either RESTRICT, SET NULL, SET DEFAULT, or CASCADE.

ON UPDATE and ON DELETE determine what happens to the foreign key when the referenced primary key is updated or deleted. When several foreign keys refer to the same primary key, different actions can be specified for each foreign key.

MySQL has several limitations on primary key updates and deletes:

- RESTRICT is applied when the ON UPDATE or ON DELETE clause is omitted.
- SET NULL cannot be used when a foreign key is not allowed NULL values.
- SET DEFAULT is not supported in some MySQL configurations.

ON UPDATE and ON DELETE are standard SQL. The clauses are supported by most relational databases, but details and limitations vary.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

PARTICIPATION ACTIVITY 35.11.8: Foreign key constraints with ON UPDATE and ON DELETE clauses.

Code	Name	ManagerID
44	Engineering	NULL
82	Sales	6381
12	Marketing	6381
~~99~~	~~Technical Support~~	~~7343~~

ID	Name	BirthDate	Salary
8754	Lisa Ellison	1993-10-02	45000
5384	Sam Snead	1995-03-15	30500
6381	Maria Rodriguez	2001-12-21	92300
~~7343~~	~~Gary Smith~~	~~1984-09-22~~	~~85000~~

```
CREATE TABLE Department (
   Code       TINYINT UNSIGNED,
   Name       VARCHAR(20),
   ManagerID SMALLINT UNSIGNED,
   PRIMARY KEY (Code),
   FOREIGN KEY (ManagerID) REFERENCES Employee(ID)
      ON DELETE CASCADE
      ON UPDATE SET NULL
);
```

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Animation content:

Static figure:
The Department table has columns Code, Name, and ManagerID. Code is the primary key. Department has four rows:
44, Engineering, 2538
82, Sales, 6381
12, Marketing, 6381
99, Technical Support, 7343

The Employee table has columns ID, Name, BirthDate, and Salary. ID is the primary key. Employee has four rows:
2538, LIsa Ellison, 1993-10-02, 45000
5384, Sam Snead, 1995-03-15, 30500
6381, Maria Rodriguez, 2001-12-21, 92300
7343, Gary Smith, 1984-09-22, 85000

An arrow points from ManagerID to ID.

An SQL statement appears:
Begin SQL code:

```
CREATE TABLE Department (
  Code TINYINT UNSIGNED,
  Name VARCHAR(20),
  ManagerID SMALLINT UNSIGNED,
  PRIMARY KEY (Code),
  FOREIGN KEY (ManagerID) REFERENCES Employee(ID)
    ON DELETE CASCADE
    ON UPDATE SET NULL
```

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

);
End SQL code.

Step 1: ManagerID is a foreign key that references the Employee ID column. The FOREIGN KEY clause is highlighted. The arrow from ManagerID to ID appears.

Step 2: ON DELETE CASCADE causes the database to delete the row with ManagerID 7343 when the employee with ID 7343 is deleted. The ON DELETE clause is highlighted. A line strikes out the fourth row of Employee:
7343, Gary Smith, 1984-09-22, 85000
A line strikes out the fourth row of Department:
99, Technical Support, 7343

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Step 3: ON UPDATE SET NULL causes the database to set ManagerID 2538 to NULL when the Employee ID 2538 is changed to 8754. The ON UPDATE clause is highlighted. In the ID column of Employee, the value 2538 changes to 8754. In the ManagerID column of Department, the value 2538 changes to NULL.

Animation captions:

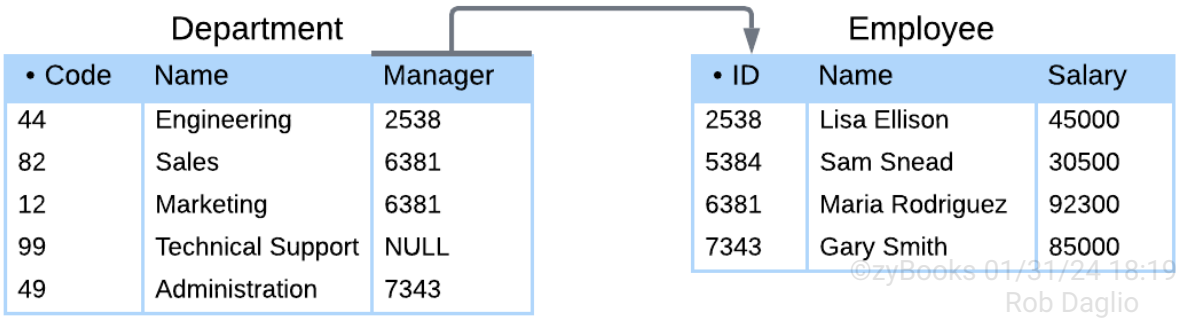
1. ManagerID is a foreign key that references the Employee ID column.
2. ON DELETE CASCADE causes the database to delete the row with ManagerID 7343 when the employee with ID 7343 is deleted.
3. ON UPDATE SET NULL causes the database to set ManagerID 2538 to NULL when the

PARTICIPATION ACTIVITY 35.11.9: ON UPDATE and ON DELETE clauses.

Refer to the table definition and data below.

```
CREATE TABLE Department (
   Code TINYINT UNSIGNED,
   Name VARCHAR(20),
   ManagerID SMALLINT UNSIGNED,
   PRIMARY KEY (Code),
   FOREIGN KEY (ManagerID) REFERENCES Employee(ID)
      ON DELETE SET NULL
      ON UPDATE CASCADE
);
```

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024



Department

• Code	Name	Manager
44	Engineering	2538
82	Sales	6381
12	Marketing	6381
99	Technical Support	NULL
49	Administration	7343

Employee

• ID	Name	Salary
2538	Lisa Ellison	45000
5384	Sam Snead	30500
6381	Maria Rodriguez	92300
7343	Gary Smith	85000

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

What is the result of each operation?

1) Delete Lisa Ellison.

- Lisa Ellison is deleted.
- Lisa Ellison is deleted, and the Engineering ManagerID is set to NULL.
- The delete is rejected.

2) Update Lisa Ellison's ID to 1000.

- Lisa Ellison's ID is set to 1000.
- Lisa Ellison's ID and the Engineering ManagerID are set to 1000.
- The update is rejected.

3) Update the Engineering ManagerID to 9999.

- The Engineering ManagerID is set to 9999.
- Lisa Ellison's ID and the Engineering ManagerID are set to 9999.
- The update is rejected.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

4) Delete Engineering.

- Engineering is deleted.
- Engineering is deleted, and Lisa Ellison's ID is set to NULL.
- The delete is rejected.

CHALLENGE ACTIVITY | 35.11.1: Referential integrity.

539740.3879454.qx3zqy7

Start

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Country

● TLD	Name	IndepDate	IndepYear
.fr	France	NULL	NULL
.jm	Jamaica	1962-08-06	1962
.mx	Mexico	1810-09-16	1810

Geography

● ISOCode3	ID	Population	15to64PopPct
FRA	.fr	66977107	0.622
JAM	.jm	2934855	0.676
MEX	.mx	126190788	0.665

With RESTRICT referential integrity, what happens if the TLD of France is updated to 'XX'?

Select

1 2 3 4 5

Check Next

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

35.12 Constraints

Column and table constraints

A ***constraint*** is a rule that governs allowable values in a database. Constraints are based on relational and business rules, and implemented with special keywords in a CREATE TABLE statement. The database automatically rejects insert, update, and delete statements that violate a constraint.

The following constraints are described elsewhere in this material:

- NOT NULL
- DEFAULT
- PRIMARY KEY
- FOREIGN KEY

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

A ***column constraint*** appears after the column name and data type in a CREATE TABLE statement. Column constraints govern values in a single column. Ex: NOT NULL is a column constraint.

A ***table constraint*** appears in a separate clause of a CREATE TABLE statement and governs values in one or more columns. Ex: FOREIGN KEY is a table constraint.

Some constraint types can be defined as either column or table constraints. Ex: A PRIMARY KEY constraint on a single column can appear either in the column declaration or a separate CREATE TABLE clause. A PRIMARY KEY constraint on a composite column must be defined as a table constraint.

Figure 35.12.1: Example constraints.

```
CREATE TABLE Employee (
   ID             INT,
   Name           VARCHAR(20) NOT NULL,
   DepartmentCode INT DEFAULT 999,
   PRIMARY KEY (ID),
   FOREIGN KEY (DepartmentCode) REFERENCES Department
(Code)
);
```

DEFAULT constraint

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

The DEFAULT constraint does not actually limit allowable values in a column. Instead, DEFAULT specifies a value that is inserted when a column is omitted from an INSERT statement. For this reason, DEFAULT is not always considered a constraint.

PARTICIPATION ACTIVITY 35.12.1: Column and table constraints.

Match the constraint with the type.

If unable to drag and drop, refresh the page.

NOT NULL | PRIMARY KEY | FOREIGN KEY

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

	Table constraint only
	Column constraint only
	Either column or table constraint

Reset

UNIQUE constraint

The ***UNIQUE*** constraint ensures that values in a column, or group of columns, are unique. When applied to a single column, UNIQUE may appear either in the column declaration or a separate clause. When applied to a group of columns, UNIQUE is a table constraint and must appear in a separate clause.

The UNIQUE constraint can be applied to primary key columns but is unnecessary, since primary keys are automatically unique.

MySQL creates an index for each UNIQUE constraint. The index stores the values of the unique column, or group of columns, in sorted order. When new values are inserted or updated, MySQL searches the index to quickly determine if the new value is unique.

PARTICIPATION ACTIVITY 35.12.2: UNIQUE constraints on the Employee table.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

```
CREATE TABLE Employee (
   ID          SMALLINT UNSIGNED,
   Name        VARCHAR(60),
   Extension   CHAR(4),
   Username    VARCHAR(50) UNIQUE,
   UNIQUE (Name, Extension),
   PRIMARY KEY (ID)
);
```

Employee

ID	Name	Extension	Username
6381	Maria Rodriguez	5050	mrodriguez
~~2538~~	~~Marty Rodriguez~~	~~5102~~	~~mrodriguez~~
5384	Maria Rodriguez	4888	mrod2
~~9001~~	~~Maria Rodriguez~~	~~5050~~	~~mariarod~~

Animation content:

Static figure:
This SQL statement appears:
Begin SQL code:

```
CREATE TABLE Employee (
  ID SMALLINT UNSIGNED,
  Name VARCHAR(60),
  Extension CHAR(4),
  Username VARCHAR(50) UNIQUE,
  UNIQUE (Name, Extension),
  PRIMARY KEY (ID)
);
```

End SQL code.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

The Employee table appears with columns ID, Name, Extension, and Username. ID is the primary key. Employee has four rows:
6381, Maria Rodriguez, 5050, mrodriguez
2538, Marty Rodriguez, 5102, mrdoriguez
5384, Maria Rodriguez, 4888, mrod2
9001, Maria Rodriguez, 5050, mariarod
A line strikes through the rows two and four.

Step 1: Username has a UNIQUE column constraint. Each Username value must be different. In the clause Username VARCHAR(50) UNIQUE, the UNIQUE keyword is highlighted. The Employee table appears with no rows.

Step 2: Attempting to insert a second row with name 'mrodriguez' fails because 'mrodriguez' already exists. The first two rows are added to Employee. A line strikes through row two.

Step 3: The UNIQUE table constraint requires each combination of Name and Extension values be different. The UNIQUE (Name, Extension) clause is highlighted.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Step 4: Inserting Maria Rodriquez with extension 4888 does not violate the UNIQUE table constraint because (Maria Rodriguez, 5050) and (Maria Rodriguez, 4888) are different. Row three is added to Employee.

Step 5: Attempting to insert Maria Rodriquez with extension 5050 fails because (Maria Rodriguez, 5050) already exists. Row four is added to Employee. A line strikes through row four.

Animation captions:

1. Username has a UNIQUE column constraint. Each Username value must be different.
2. Attempting to insert a second row with name 'mrodriguez' fails because 'mrodriguez' already exists.
3. The UNIQUE table constraint requires each combination of Name and Extension values be different.
4. Inserting Maria Rodriquez with extension 4888 does not violate the UNIQUE table constraint because (Maria Rodriguez, 5050) and (Maria Rodriguez, 4888) are different.
5. Attempting to insert Maria Rodriquez with extension 5050 fails because (Maria Rodriguez, 5050) already exists.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

PARTICIPATION ACTIVITY 35.12.3: UNIQUE constraint.

Indicate whether the rows violate the UNIQUE constraints on Department.

```
CREATE TABLE Department (
   Code        TINYINT UNSIGNED,
   Name        VARCHAR(20) UNIQUE,
   ManagerID   SMALLINT,
   Appointment DATE,
   PRIMARY KEY (Code),
   UNIQUE (ManagerID, Appointment)
);
```

1)
```
(44, 'Engineering', 2538,
'2020-01-15')
(82, 'Sales', 6381, '2019-08-
01')
```

- OK
- Violates

2)
```
(12, 'Marketing', 2538, '2019-
11-22')
(99, 'Marketing', NULL, NULL)
```

- OK
- Violates

3)
```
(44, 'Engineering', 2538,
'2020-01-15')
(82, 'Sales', 2538, '2020-01-
15')
```

- OK
- Violates

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

4)
```
(12, 'Marketing', 5384, '2019-
11-22')
(99, 'Technical Support', 5384,
NULL)
```

- OK
- Violates

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

CHECK constraint

The ***CHECK*** constraint specifies an expression on one or more columns of a table. The constraint is violated when the expression is FALSE and satisfied when the expression is either TRUE or NULL.

When the expression contains one column, CHECK may appear either in the column declaration or a separate clause. When the expression contains multiple columns, CHECK is a table constraint and must be a separate clause.

PARTICIPATION ACTIVITY 35.12.4: CHECK constraint.

```
CREATE TABLE Employee (
   ID        SMALLINT UNSIGNED,
   Name      VARCHAR(60),
   BirthDate DATE,
   HireDate  DATE CHECK (HireDate >= '2000-01-01' AND HireDate <= '2019-12-31'),
   CHECK (BirthDate < HireDate),
   PRIMARY KEY (ID)
);
```

Employee

ID	Name	BirthDate	HireDate
6381	Maria Rodriguez	1970-12-01	2000-01-31
~~2538~~	~~Lisa Ellison~~	~~2000-01-01~~	~~2020-03-15~~
~~5384~~	~~Sam Snead~~	~~2003-11-29~~	~~2003-11-01~~
8312	Jiho Chen	NULL	2013-06-03

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Animation content:

Static figure:
An SQL statement appears:
Begin SQL code:
CREATE TABLE Employee (

```
  ID SMALLINT UNSIGNED,
  Name VARCHAR(60),
  BirthDate DATE,
  HireDate DATE CHECK (HireDate >= '2000-01-01' AND HireDate <= '2019-12-31'),
  CHECK (BirthDate < HireDate),
  PRIMARY KEY (ID)
);
```

End SQL code.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

The Employee table appears with columns ID, Name, BirthDate, and HireDate. ID is the primary key. Employee has four rows:
6381, Maria Rodriguez, 1970-12-01, 2000-01-31
2538, Lisa Ellison, 2000-01-01, 2020-03-15
5384, Sam Snead, 2003-11-29, 2003-11-01
8312, Jiho Chen, NULL, 2013-06-03
A line strikes out rows two and three.

Step 1: The CHECK column constraint ensures all rows have a HireDate between Jan 1, 2000 and Dec 31, 2019. The clause CHECK (HireDate >= '2000-01-01' AND HireDate <= '2019-12-31') is highlighted. The Employee table appears with no rows.

Step 2: Maria's HireDate is between Jan 1, 2000 and Dec 31, 2019. But inserting Lisa fails because Mar 15, 2020 is after Dec 31, 2019. The first two rows are added to Employee. A line strikes out row two.

Step 3: The CHECK table constraint states BirthDate must precede HireDate. The clause CHECK (BirthDate < HireDate) is highlighted.

Step 4: Inserting Sam fails because the BirthDate Nov 29, 2003 is after HireDate Nov 1, 2003. Row three is added to Employee. A line strikes out this row.

Step 5: Jiho's birth date is NULL, so the CHECK table constraint evaluates to NULL and is not violated. Row four is added to Employee.

Animation captions:

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

1. The CHECK column constraint ensures all rows have a HireDate between Jan 1, 2000 and Dec 31, 2019.
2. Maria's HireDate is between Jan 1, 2000 and Dec 31, 2019. But inserting Lisa fails because Mar 15, 2020 is after Dec 31, 2019.
3. The CHECK table constraint states BirthDate must precede HireDate.
4. Inserting Sam fails because the BirthDate Nov 29, 2003 is after HireDate Nov 1, 2003.
5. Jiho's birth date is NULL, so the CHECK table constraint evaluates to NULL and is not violated.

PARTICIPATION ACTIVITY 35.12.5: CHECK constraint.

Indicate whether each row violates the CHECK constraints on Department. The Size column constraint limits values to 'small', 'medium', or 'large'.

```
CREATE TABLE Department (
   Code          TINYINT UNSIGNED,
   Name          VARCHAR(20),
   ManagerID     SMALLINT,
   AdminAssistID SMALLINT,
   Size          VARCHAR(6) CHECK (Size IN ('small', 'medium', 'large')),
   PRIMARY KEY (Code),
   CHECK (ManagerID >= 1000 AND ManagerID <> AdminAssistID)
);
```

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

1) `(44, 'Engineering', 2538, 1024, 'tiny')`

   - OK
   - Violates

2) `(82, 'Sales', 999, 1024, 'medium')`

   - OK
   - Violates

3) `(99, 'Technical Support', 5384, 5384, 'large')`

   - OK
   - Violates

4) `(99, 'Technical Support', 5384, 2399, 'large')`

   - OK
   - Violates

5) `(50, 'Maintenance', NULL, 4380, NULL)`

   - OK
   - Violates

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Constraint names

Table constraints may be named using the optional ***CONSTRAINT*** keyword, followed by the constraint name and declaration. If no name is provided, the database generates a default name. Constraint names appear in error messages when constraints are violated.

Most column constraints cannot be named. However, the CHECK column constraint is an exception and can be named with a CONSTRAINT clause in the column declaration.

PARTICIPATION ACTIVITY 35.12.6: Constraint names.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

```
CREATE TABLE Employee (
   ID             INT,
   Name           VARCHAR(20) NOT NULL,
   DepartmentCode INT DEFAULT 999,

   CONSTRAINT EmployeePK PRIMARY KEY (ID),
); CONSTRAINT EmployeeDepartmentFK  FOREIGN KEY (DepartmentCode) REFERENCES Departm
                                   (Code)
```

Animation content:

Static figure:
An SQL statement appears:
Begin SQL code:
CREATE TABLE Employee (
  ID INT,
  Name VARCHAR(20) NOT NULL,
  DepartmentCode INT DEFAULT 999,
  CONSTRAINT EmployeePK PRIMARY KEY (ID),
  CONSTRAINT EmployeeDepartmentFK FOREIGN KEY (DepartmentCode) REFERENCES Department (Code)
);
End SQL code.

Step 1: The column constraints NOT NULL and DEFAULT cannot be named. The statement appears without CONSTRAINT EmployeePK and CONSTRAINT EmployeeDepartmentFK. The NOT NULL and the DEFAULT clauses are highlighted.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Step 2: The PRIMARY KEY table constraint is named EmployeePK with an optional CONSTRAINT clause. CONSTRAINT EmployeePK is inserted before the PRIMARY KEY clause.

Step 3: The FOREIGN KEY table constraint is named EmployeeDepartmentFK. CONSTRAINT EmployeeDepartmentFK is inserted before the FOREIGN KEY clause.

Animation captions:

1. The column constraints NOT NULL and DEFAULT cannot be named.
2. The PRIMARY KEY table constraint is named EmployeePK with an optional CONSTRAINT clause.
3. The FOREIGN KEY table constraint is named EmployeeDepartmentFK.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

MySQL constraint names

The MySQL statement ***`SHOW CREATE TABLE TableName`*** *returns the CREATE TABLE statement for TableName. The statement shows all CONSTRAINT ConstraintName clauses but does not show default constraint names.*

The following query displays all names of constraints on TableName, including default names:

```
SELECT Column_Name, Constraint_Name
FROM Information_Schema.Key_Column_Usage
WHERE Table_Name = 'TableName';
```

Key_Column_Usage is a table from the Information_Schema system database. Other Key_Column_Usage columns contain additional constraint details. Ex: Referenced_Table_Name and Referenced_Column_Name provide information about foreign key constraints.

PARTICIPATION ACTIVITY 35.12.7: Constraint names.

1) Name the primary key table constraint 'DepartmentKey'.

```
CREATE TABLE Department (
   Code INT,
   Name VARCHAR(20),
   ManagerID INT,
   
PRIMARY KEY (Code)
);
```

Check Show answer

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

2) Name the CHECK column constraint 'CheckManager'

```
CREATE TABLE Department (
   Code INT,
   Name VARCHAR(20),
   ManagerID INT

                    CHECK
(ManagerID > 999),
   PRIMARY KEY (Code)
);
```

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Check Show answer

3) Add a UNIQUE constraint called UniqueNameMgr that ensures the Name and ManagerID combination are unique.

```
CREATE TABLE Department (
   Code TINYINT UNSIGNED,
   Name VARCHAR(20),
   ManagerID SMALLINT,


,
   PRIMARY KEY (Code)
);
```

Check Show answer

Adding and dropping constraints

Constraints are added and dropped with the `ALTER TABLE TableName` followed by an ADD, DROP, or CHANGE clause.

Unnamed constraints such as NOT NULL and DEFAULT are added or dropped with a CHANGE clause:

- `CHANGE CurrentColumnName NewColumnName NewDataType [ConstraintDeclarat`

Named constraints are added with an ADD clause:

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

- `ADD [CONSTRAINT ConstraintName] PRIMARY KEY (Column1, Column2 ...)`
- `ADD [CONSTRAINT ConstraintName] FOREIGN KEY (Column1, Column2 ...) REF`
- `ADD [CONSTRAINT ConstraintName] UNIQUE (Column1, Column2 ...)`
- `ADD [CONSTRAINT ConstraintName] CHECK (expression)`

Adding a constraint fails when the table contains data that violates the constraint.

Named constraints are dropped with a DROP clause:

- `DROP PRIMARY KEY`
- `DROP FOREIGN KEY ConstraintName`
- `DROP INDEX ConstraintName` (drops UNIQUE constraints)
- `DROP CHECK ConstraintName`
- `DROP CONSTRAINT ConstraintName` (drops any named constraint)

Dropping a table fails when a foreign key constraint refers to the table's primary key. Before dropping the table, either the foreign key constraint or the foreign key table must be dropped.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

PARTICIPATION ACTIVITY 35.12.8: Adding and dropping constraints.

```
CREATE TABLE Employee (
   ID         SMALLINT UNSIGNED,
   Name       VARCHAR(60),
   HireDate   DATE CONSTRAINT HireCheck CHECK (HireDate >= '2000-01-01'),
   CONSTRAINT UniqueNameHiredate UNIQUE (Name, HireDate),
   PRIMARY KEY (ID)
);
```

Name: HireCh

Name: UniqueNameHiredate

Add a new row: (305, 'Fred Yu', '1999-12-30') ✗

Error message: Check on constraint 'HireCheck' is violated.

```
ALTER TABLE Employee
DROP CHECK HireCheck;
```

Add a new row: (305, 'Fred Yu', '1999-12-30') ✓

```
ALTER TABLE Employee
ADD CONSTRAINT HireCheck CHECK (HireDate < '2000-02-14');
```

Add a new row: (610, 'Emma Jackson', '2014-12-30') ✗

Error message: Check on constraint 'HireCheck' is violated.

Animation content:

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Static figure:
Three SQL statements appear:
Begin SQL code:
CREATE TABLE Employee (
ID SMALLINT UNSIGNED,
Name VARCHAR(60),
HireDate DATE CONSTRAINT HireCheck CHECK (HireDate >= '2000-01-01'),
CONSTRAINT UniqueNameHiredate UNIQUE (Name, HireDate),

```
   PRIMARY KEY (ID)
);

ALTER TABLE Employee
DROP CHECK HireCheck;

ALTER TABLE Employee
ADD CONSTRAINT HireCheck CHECK (HireDate < '2002-02-14');;
End SQL code.
```

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Step 1: The CONSTRAINT keyword gives the CHECK and UNIQUE constraints user-defined names. The first statement appears. The two CONSTRAINT clauses are highlighted.

Step 2: The database uses constraint names in error messages. Fred's HireDate is not >= 2000-01-01. A row appears below the first statement with caption 'Add new row':
305, 'Fred Yu', '1999-12-30'
The error message"Check on constraint 'HireCheck' is violated" appears. An X appears next to the row.

Step 3: The ALTER TABLE statement drops the HireCheck constraint. Any HireDate may now be added to Employee. The second statement appears. A row appears below the second statement with caption 'Add new row':
305, 'Fred Yu', '1999-12-30'
A check appears next to the row.

Step 4: The ALTER TABLE statement adds a new constraint that ensures the HireDate is before Feb 14, 2000. Emma's HireDate violates the new constraint. The third statement appears. A row appears below the third statement with caption 'Add new row':
610, 'Emma Jackson', '2014-12-30'
The error message"Check on constraint 'HireCheck' is violated" appears. An X appears next to the row.

Animation captions:

1. The CONSTRAINT keyword gives the CHECK and UNIQUE constraints user-defined names.
2. The database uses constraint names in error messages. Fred's HireDate is not >= 2000-01-01.
3. The ALTER TABLE statement drops the HireCheck constraint. Any HireDate may now be added to Employee.
4. The ALTER TABLE statement adds a new constraint that ensures the HireDate is before Feb

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

PARTICIPATION ACTIVITY 35.12.9: Adding and dropping constraints.

1) Add a NOT NULL constraint to an existing column Salary in the Department table.

```
ALTER TABLE Department
[          ] Salary INT
NOT NULL;
```

Check Show answer

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

2) Add a constraint called MgrIDCheck to the existing Department table to ensure ManagerID is 2000 or above.

```
ALTER TABLE Department
[                    ]
(ManagerID >= 2000);
```

Check Show answer

3) Drop the UNIQUE constraint called UniqueNameMgr from the existing Department table.

```
ALTER TABLE Department
[                    ]
;
```

Check Show answer

CHALLENGE ACTIVITY 35.12.1: Constraints.

539740.3879454.qx3zqy7

Start

```
CREATE TABLE Country (
    ISOCode2 CHAR(2),
    IndepYear SMALLINT,
    Name VARCHAR(15),
    ISOCode3 CHAR(3) UNIQUE,
    UNIQUE (Name, IndepYear),
    PRIMARY KEY (ISOCode2)
);
```

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Indicate whether each pair of rows violate the Country table's UNIQUE constraints.

Select

ISOCode2	IndepYear	Name	ISOCode3
AA	52	Valhalla	ALL
LL	52	Valhalla	AHA

Select

ISOCode2	IndepYear	Name	ISOCode3
AV	41	Valhalla	VLL
AL	7	Valhalla	VLL

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

1 2 3

Check Next

Exploring further:

- MySQL constraint handling
- MySQL CHECK constraint
- MySQL ALTER TABLE statement

35.13 LAB - Create Movie table

Create a **Movie** table with the following columns:

- **ID** - positive integer with maximum value of 50,000
- **Title** - variable-length string with up to 50 characters
- **Rating** - fixed-length string with 4 characters
- **ReleaseDate** - date
- **Budget** - decimal value representing a cost of up to 999,999 dollars, with 2 digits for cents

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Note: Your SQL code does not display any results in Develop mode. Use Submit mode to test your code.

539740.3879454.qx3zqy7

LAB ACTIVITY | 35.13.1: LAB - Create Movie table | 0 / 10

Main.sql Load default template...

```
1 -- Your SQL statement goes here
```

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Develop mode | Submit mode

Explore the database and run your program as often as you'd like, before submitting for grading. Click **Run program** and observe the program's output in the second box.

Run program

Main.sql (Your program) → Output (shown below)

Program output displayed here

Coding trail of your work What is this?

```
History of your effort will appear here once you begin working
on this zyLab.
```

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

35.14 LAB - Alter Movie table

The **Movie** table has the following columns:

- **ID** - positive integer
- **Title** - variable-length string

- **Genre** - variable-length string
- **RatingCode** - variable-length string
- **Year** - integer

Write ALTER statements to make the following modifications to the Movie table:

1. Add a **Producer** column with VARCHAR data type (max 50 chars).
2. Remove the **Genre** column.
3. Change the Year column's name to **ReleaseYear**, and change the data type to **SMALLINT**.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Note: Your SQL code does not display any results in Develop mode. Use Submit mode to test your code.

539740.3879454.qx3zqy7

LAB ACTIVITY | 35.14.1: LAB - Alter Movie table | 0 / 10

Main.sql — Load default template...

```
1 -- Your SQL statements go here
2
```

Develop mode | **Submit mode**

Explore the database and run your program as often as you'd like, before submitting for grading. Click **Run program** and observe the program's output in the second box.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Run program

Main.sql (Your program) → Output (shown below)

Program output displayed here

Coding trail of your work What is this?

```
History of your effort will appear here once you begin working
on this zyLab.
```

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

35.15 LAB - Select horses with logical operators

The **Horse** table has the following columns:

- **ID** - integer, primary key
- **RegisteredName** - variable-length string
- **Breed** - variable-length string
- **Height** - decimal number
- **BirthDate** - date

Write a SELECT statement to select the registered name, height, and birth date for only horses that have a height between 15.0 and 16.0 (inclusive) or have a birth date on or after January 1, 2020.

539740.3879454.qx3zqy7

LAB ACTIVITY | 35.15.1: LAB - Select horses with logical operators | 0 / 10

Main.sql Load default template...

```
1 -- Your SELECT statement goes here
2
```

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Develop mode | Submit mode

Explore the database and run your program as often as you'd like, before submitting for grading. Click **Run program**

and observe the program's output in the second box.

Run program

Main.sql
(Your program) → Output (shown below)

Program output displayed here

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Coding trail of your work What is this?

```
History of your effort will appear here once you begin working
on this zyLab.
```

35.16 LAB - Insert rows into Horse table

The **Horse** table has the following columns:

- **ID** - integer, auto increment, primary key
- **RegisteredName** - variable-length string
- **Breed** - variable-length string, must be one of the following: Egyptian Arab, Holsteiner, Quarter Horse, Paint, Saddlebred
- **Height** - decimal number, must be between 10.0 and 20.0
- **BirthDate** - date, must be on or after Jan 1, 2015

Insert the following data into the Horse table:

RegisteredName	Breed	Height	BirthDate
Babe	Quarter Horse	15.3	2015-02-10
Independence	Holsteiner	16.0	2017-03-13
Ellie	Saddlebred	15.0	2016-12-22
NULL	Egyptian Arab	14.9	2019-10-12

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Notes:

- Your SQL code does not display any results in Develop mode. Use Submit mode to test your code.

- In another lab that creates the Horse table, RegisteredName is NOT NULL. In this lab, NULL is allowed in this column.

539740.3879454.qx3zqy7

LAB ACTIVITY | 35.16.1: LAB - Insert rows into Horse table | 0 / 10

Main.sql

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Load default template...

```
1 -- Your SQL statements goes here
2
```

Develop mode | **Submit mode**

Explore the database and run your program as often as you'd like, before submitting for grading. Click **Run program** and observe the program's output in the second box.

Run program

Main.sql (Your program) → Output (shown below)

Program output displayed here

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Coding trail of your work What is this?

History of your effort will appear here once you begin working on this zyLab.

35.17 LAB - Update rows in Horse table

The **Horse** table has the following columns:

- **ID** - integer, auto increment, primary key
- **RegisteredName** - variable-length string
- **Breed** - variable-length string, must be one of the following: Egyptian Arab, Holsteiner, Quarter Horse, Paint, Saddlebred
- **Height** - decimal number, must be ≥ 10.0 and ≤ 20.0
- **BirthDate** - date, must be ≥ Jan 1, 2015

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Make the following updates:

1. Change the height to **15.6** for horse with ID 2.
2. Change the registered name to **Lady Luck** and birth date to **May 1, 2015** for horse with ID 4.
3. Change every horse breed to ***NULL*** for horses born on or after December 22, 2016.

Note: Your SQL code does not display any results in Develop mode. Use Submit mode to test your code.

539740.3879454.qx3zqy7

LAB ACTIVITY | 35.17.1: LAB - Update rows in Horse table | 0 / 10

Main.sql — Load default template...

```
1 -- Your SQL statements goes here
2
```

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Develop mode | **Submit mode**

Explore the database and run your program as often as you'd like, before submitting for grading. Click **Run program** and observe the program's output in the second box.

Run program

Main.sql
(Your program)

→ Output (shown below)

Program output displayed here

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Coding trail of your work What is this?

```
History of your effort will appear here once you begin working
on this zyLab.
```

35.18 LAB - Delete rows from Horse table

The **Horse** table has the following columns:

- **ID** - integer, auto increment, primary key
- **RegisteredName** - variable-length string
- **Breed** - variable-length string
- **Height** - decimal number
- **BirthDate** - date

Delete the following rows:

1. Horse with ID 5.
2. All horses with breed Holsteiner or Paint.
3. All horses born before March 13, 2013.

Note: Your SQL code does not display any results in Develop mode. Use Submit mode to test your code.

539740.3879454.qx3zqy7

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

LAB ACTIVITY | 35.18.1: LAB - Delete rows from Horse table 0 / 10

Main.sql Load default template...

```
1 -- Your SQL statements goes here
2
```

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Develop mode | Submit mode

Explore the database and run your program as often as you'd like, before submitting for grading. Click **Run program** and observe the program's output in the second box.

Run program

Main.sql
(Your program) → Output (shown below)

Program output displayed here

Coding trail of your work What is this?

```
History of your effort will appear here once you begin working
on this zyLab.
```

35.19 LAB - Create Horse table with constraints

Create a **Horse** table with the following columns, data types, and constraints. NULL is allowed unless 'not NULL' is explicitly stated.

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

- **ID** - integer with range 0 to 65535, auto increment, primary key
- **RegisteredName** - variable-length string with max 15 chars, not NULL
- **Breed** - variable-length string with max 20 chars, must be one of the following: Egyptian Arab, Holsteiner, Quarter Horse, Paint, Saddlebred
- **Height** - number with 3 significant digits and 1 decimal place, must be ≥ 10.0 and ≤ 20.0

- **BirthDate** - date, must be ≥ Jan 1, 2015

Notes: Not all constraints can be tested due to current limitations of MySQL. Your SQL code does not display any results in Develop mode. Use Submit mode to test your code.

539740.3879454.qx3zqy7

LAB ACTIVITY | 35.19.1: LAB - Create Horse table with constraints 0 / 10

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Main.sql

Load default template...

```
1 -- Your SQL statement goes here
2
```

Develop mode | Submit mode

Explore the database and run your program as often as you'd like, before submitting for grading. Click **Run program** and observe the program's output in the second box.

Run program

Main.sql (Your program) → Output (shown below)

Program output displayed here

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Coding trail of your work What is this?

History of your effort will appear here once you begin working on this zyLab.

35.20 LAB - Create Student table with constraints

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Create a **Student** table with the following column names, data types, and constraints:

- **ID** - integer with range 0 to 65 thousand, auto increment, primary key
- **FirstName** - variable-length string with max 20 chars, not NULL
- **LastName** - variable-length string with max 30 chars, not NULL
- **Street** - variable-length string with max 50 chars, not NULL
- **City** - variable-length string with max 20 chars, not NULL
- **State** - fixed-length string of 2 chars, not NULL, default "TX"
- **Zip** - integer with range 0 to 16 million, not NULL
- **Phone** - fixed-length string of 10 chars, not NULL
- **Email** - variable-length string with max 30 chars, must be unique

Note: Your SQL code does not display any results in Develop mode. Use Submit mode to test your code.

539740.3879454.qx3zqy7

LAB ACTIVITY | 35.20.1: LAB - Create Student table with constraints | 0 / 10

Main.sql — Load default template...

```
1 -- Your SQL statements go here
2
```

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Develop mode | Submit mode

Explore the database and run your program as often as you'd like, before submitting for grading. Click **Run program** and observe the program's output in the second box.

Run program

Main.sql
(Your program)

→ Output (shown below)

©zyBooks 01/31/24 18:19 1939727
MDCCOP2335Spring2024

Program output displayed here

Coding trail of your work What is this?

```
History of your effort will appear here once you begin working on this zyLab.
```

35.21 LAB - Create LessonSchedule table with FK constraints

The database contains a **Horse** table, with columns:

- **ID** - integer, primary key
- **RegisteredName** - variable-length string

The database contains a **Student** table, with columns:

- **ID** - integer, primary key
- **FirstName** - variable-length string
- **LastName** - variable-length string

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Create a third table, named **LessonSchedule**, with columns:

- **HorseID** - integer with range 0 to 65 thousand, not NULL, foreign key references Horse(ID)
- **StudentID** - integer with range 0 to 65 thousand, foreign key references Student(ID)
- **LessonDateTime** - date/time, not NULL
- Primary key is **(HorseID, LessonDateTime)**

If a row is deleted from Horse, the rows with the same horse ID should be deleted from LessonSchedule automatically.

If a row is deleted from Student, the same student IDs should be set to NULL in LessonSchedule automatically.

Notes: Table and column names are case sensitive in the auto-grader. Your SQL code does not display any results in Develop mode. Use Submit mode to test your code.

539740.3879454.qx3zqy7

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

LAB ACTIVITY 35.21.1: LAB - Create LessonSchedule table with FK constraints 0 / 10

Main.sql Load default template...

```
1  CREATE TABLE Horse (
2     ID               SMALLINT UNSIGNED AUTO_INCREMENT,
3     RegisteredName   VARCHAR(15),
4     PRIMARY KEY (ID)
5  );
6
7  CREATE TABLE Student (
8     ID               SMALLINT UNSIGNED AUTO_INCREMENT,
9     FirstName        VARCHAR(20),
10    LastName         VARCHAR(30),
11    PRIMARY KEY (ID)
12 );
13
14 -- Your SQL statements go here
15
```

Develop mode | **Submit mode**

Explore the database and run your program as often as you'd like, before submitting for grading. Click **Run program** and observe the program's output in the second box.

Run program

Main.sql (Your program) → Output (shown below)

©zyBooks 01/31/24 18:19 1939727
Rob Daglio
MDCCOP2335Spring2024

Program output displayed here

Coding trail of your work What is this?