

SQL CRUD

Create, Read, Update, Delete

CRUD Operations in SQL

- ▶ Create (INSERT)
- ▶ Read (SELECT - covered later)
- ▶ Update (UPDATE)
- ▶ Delete (DELETE, DROP)

INSERT Command

General form is

```
INSERT INTO <table_name> (<column_name> [, ...]) VALUES  
    (<new_value> [, ...]);
```

Example:

```
insert into author (author_id, first_name, second_name)  
    values (1, "Jenny", "McCarthy");
```

Can leave off column names list to insert positionally:

```
insert into author values (1, "Jenny", "McCarthy");
```

UPDATE Command

General form:

```
UPDATE <table_name> SET <column_name>=<new_value> [, ...]  
    WHERE expression
```

Example: Surely we meant Lisp inventor, AI co-founder, and Turing Award winner John McCarthy instead of anti-vaxxer Jenny McCarthy.

```
update author set first_name='John' where  
    last_name='McCarthy';
```

Notice that we can use single or double quotes in most RDBMS systems.

DELETE Command

General form:

```
DELETE FROM <table_name> WHERE <boolean_expression>;
```

Example:

```
delete from author where last_name="Batty";
```

Can also drop whole tables:

```
DROP TABLE <table_name>
```

Referential Integrity

To maintain integrity on update or delete specify:

- ▶ For ON DELETE:
 - ▶ SET NULL
 - ▶ SET DEFAULT
- ▶ For ON UPDATE
 - ▶ CASCADE

Note: for MySQL ON DELETE RESTRICT is the default.

Referential Integrity - SET NULL

Example:

```
CREATE TABLE pub (  
  pub_id INT PRIMARY KEY,  
  title VARCHAR(16) NOT NULL,  
  book_id INT,  
  foreign key (book_id) REFERENCES book(book_id) ON  
    DELETE SET NULL  
);
```

Means that if the row from the book table containing book_id is deleted, then book_id is set to NULL for each affected row in the pub table. Notice that if you choose SET NULL as your ON DELETE action your column definition must allow nulls.

Referential Integrity Constraints in MySQL

MySQL will only enforce referential integrity constraints that are specified separately from column definitions as above. The following syntax:

```
CREATE TABLE pub (  
  pub_id INT PRIMARY KEY,  
  title VARCHAR(16) NOT NULL,  
  book_id INT REFERENCES book(book_id) ON DELETE SET NULL  
);
```

is valid SQL syntax but is ignored by MySQL's default InnoDB engine.

Referential Integrity - SET DEFAULT

Example:

```
CREATE TABLE pub (  
  pub_id INT PRIMARY KEY,  
  title VARCHAR(16) NOT NULL,  
  book_id INT DEFAULT 0 REFERENCES book(book_id) ON  
    DELETE SET DEFAULT  
);
```

Means that if the row from the book table containing book_id is deleted, then book_id is set to 0 for each affected row in the pub table.

Note: MySQL's default InnoDB engine does not implement ON DELETE SET DEFAULT.

Referential Integrity - CASCADE

Example:

```
CREATE TABLE pub (  
  pub_id INT PRIMARY KEY,  
  title VARCHAR(16) NOT NULL,  
  book_id INT NOT NULL,  
  FOREIGN KEY (book_id) REFERENCES book(book_id) ON  
    UPDATE CASCADE  
);
```

Means that if a `book_id` value changes for a row in the `book` table, the change is applied to the affected rows of the `pub` table also.

Multiple Referential Integrity Constraints

You would normally set constraints for updates and deletes.

Example:

```
CREATE TABLE pub (  
  pub_id INT PRIMARY KEY,  
  title VARCHAR(16) NOT NULL,  
  book_id INT,  
  FOREIGN KEY (book_id) REFERENCES book(book_id)  
    ON UPDATE CASCADE  
    ON DELETE SET NULL  
);
```