How does SQL allow implementation of the entity integrity and referential integrity constraints described in Chapter 3? What about referential triggered actions?

PRIMARY KEY clause

FOREIGN KEY clause

Default SQL action for an integrity violation is to reject the update operation, known as REJECT (p. 186)

Referential triggered actions: SET NULL, CASCADE, and SET DEFAULT

SQL allows the implementation of the entity integrity constraint through the PRIMARY KEY clause. Take a look at the following SQL DDL to create a CUSTOMER relation:

CREATE TABLE customers (

email\_address varchar NOT NULL,

last\_name varchar,

first\_name varchar,

PRIMARY KEY (email\_address)

);

The PRIMARY KEY command declares that the email\_address column is the primary key of the database table CUSTOMERS. Declaring that email\_address is a primary key, SQL understands that the value of this attribute, a customer’s email address, in each record of the CUSTOMERS table must be unique and not null, thus supporting the entity integrity constraint in the relation.

To implement referential integrity constraints, SQL allows primary keys from one table to be referenced in another table. Take the example of an ORDERS tables below:

CREATE TABLE orders (

order\_id int NOT NULL,

email\_address varchar NOT NULL,

PRIMARY KEY (order\_id),

FOREIGN KEY (email\_address) REFERENCES customers (email\_address)

…

);

The ORDERS table references the email\_address column in the CUSTOMERS table, so that each order is linked to the customer who submitted it. Using the FOREIGN KEY and REFERENCES keywords, the value of the email\_address column for each record in ORDERS will correspond to a value in the email\_address column in CUSTOMERS. By default, any update operation that violates an integrity constraint will be rejected unless a referentially triggered action is specified.

Referentially triggered actions are alternate actions that occur when a referential integrity constraint is violated. These actions can be broken down into two parts: when the action takes place and what happens when the action takes place. The actions may take place when a primary key value is updated or deleted as specified by ON UPDATE and ON DELETE, respectively. For each of these instances, the action that takes place can be specified with RESTRICT, CASCADE, SET NULL, or SET DEFAULT. If RESTRICT is used, when the primary key is altered, the foreign key is left unchanged. Conversely, CASCADE forces the changes on the primary key to extend to any foreign key that references it. SET NULL and SET DEFAULT force the foreign key to take NULL as its value or take a default value, respectively.

References:

Aubry, F. (2024, June 19). *Integrity Constraints in SQL: A Guide with Examples*. DataCamp. Retrieved 01 30, 2025, from https://www.datacamp.com/tutorial/integrity-constraints-sql

*Documentation: 17: 5.5. Constraints*. (n.d.). PostgreSQL. Retrieved January 30, 2025, from <https://www.postgresql.org/docs/current/ddl-constraints.html#DDL-CONSTRAINTS-PRIMARY-KEYS>

Elmasri, R., & Navathe, S. (2016). *Fundamentals of Database Systems*. Pearson.

Lowhorn, J. (2025). *Week 3: Basic SQL* [PowerPoint Slides]. Canvas@Merrimack. <https://canvas.merrimack.edu/courses/15276/pages/module-03-content?module_item_id=963181>

<https://www.postgresql.org/docs/current/ddl-constraints.html#DDL-CONSTRAINTS-PRIMARY-KEYS>

https://www.datacamp.com/tutorial/integrity-constraints-sql