3.6.6 Using Foreign Keys

MySQL supports foreign keys, which permit cross-referencing related data across tables, and foreign key constraints, which help keep the related data consistent.

A foreign key relationship involves a parent table that holds the initial column values, and a child table with column values that reference the parent column values. A foreign key constraint is defined on the child table.

This following example relates `parent` and `child` tables through a single-column foreign key and shows how a foreign key constraint enforces referential integrity.

Create the parent and child tables:

```
CREATE TABLE parent (
    id INT NOT NULL,
    PRIMARY KEY (id)
) ENGINE=INNODB;


CREATE TABLE child (
    id INT,
    parent_id INT,
    INDEX par_ind (parent_id),
    FOREIGN KEY (parent_id)
        REFERENCES parent(id)
) ENGINE=INNODB;
```

Insert a row into the parent table:

```
mysql> INSERT INTO parent (id) VALUES (1);
```

Verify that the data was inserted:

```
mysql> SELECT * FROM parent;
+----+
| id |
+----+
|  1 |
+----+
```

Insert a row into the child table:

```
mysql> INSERT INTO child (id,parent_id) VALUES (1,1);
```

The insert operation is successful because `parent_id` 1 is present in the parent table.

Insert a row into the child table with a `parent_id` value that is not present in the parent table:

```
mysql> INSERT INTO child (id,parent_id) VALUES(2,2);
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails
(`test`.`child`, CONSTRAINT `child_ibfk_1` FOREIGN KEY (`parent_id`)
REFERENCES `parent` (`id`))
```

The operation fails because the specified `parent_id` value does not exist in the parent table.

Try to delete the previously inserted row from the parent table:

```
mysql> DELETE FROM parent WHERE id VALUES = 1;
ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint fails
```

```
(`test`.`child`, CONSTRAINT `child_ibfk_1` FOREIGN KEY (`parent_id`)
REFERENCES `parent` (`id`))
```

This operation fails because the record in the child table contains the referenced id (`parent_id`) value.

When an operation affects a key value in the parent table that has matching rows in the child table, the result depends on the referential action specified by `ON UPDATE` and `ON DELETE` subclauses of the `FOREIGN KEY` clause. Omitting `ON DELETE` and `ON UPDATE` clauses (as in the current child table definition) is the same as specifying the `RESTRICT` option, which rejects operations that affect a key value in the parent table that has matching rows in the parent table.

To demonstrate `ON DELETE` and `ON UPDATE` referential actions, drop the child table and recreate it to include `ON UPDATE` and `ON DELETE` subclauses with the `CASCADE` option. The `CASCADE` option automatically deletes or updates matching rows in the child table when deleting or updating rows in the parent table.

```
DROP TABLE child;

CREATE TABLE child (
    id INT,
    parent_id INT,
    INDEX par_ind (parent_id),
    FOREIGN KEY (parent_id)
        REFERENCES parent(id)
        ON UPDATE CASCADE
        ON DELETE CASCADE
) ENGINE=INNODB;
```

Insert the following rows into the child table:

```
mysql> INSERT INTO child (id,parent_id) VALUES(1,1),(2,1),(3,1);
```

Verify that the data was inserted:

```
mysql> SELECT * FROM child;
+------+-----------+
| id   | parent_id |
+------+-----------+
|    1 |         1 |
|    2 |         1 |
|    3 |         1 |
+------+-----------+
```

Update the id in the parent table, changing it from 1 to 2.

```
mysql> UPDATE parent SET id = 2 WHERE id = 1;
```

Verify that the update was successful:

```
mysql> SELECT * FROM parent;
+----+
| id |
+----+
|  2 |
+----+
```

Verify that the `ON UPDATE CASCADE` referential action updated the child table:

```
mysql> SELECT * FROM child;
+------+-----------+
| id   | parent_id |
```

```
+------+-----------+
|    1 |         2 |
|    2 |         2 |
|    3 |         2 |
+------+-----------+
```

To demonstrate the `ON DELTE CASCADE` referential action, delete records from the parent table where the `parent_id = 2`, which deletes all records in the parent table.

```
mysql> DELETE FROM parent WHERE id = 2;
```

Because all records in the child table are associated with `parent_id = 2`, the `ON DELETE CASCADE` referential action removes all records from the child table:

```
mysql> SELECT * FROM child;
Empty set (0.00 sec)
```

For more information about foreign key constraints, see Section 13.1.20.5, “FOREIGN KEY Constraints”.