12.2.9.6. EXISTS and NOT EXISTS

If a subquery returns any rows at all, **EXISTS** *subquery* is **TRUE**, and **NOT EXISTS** *subquery* is **FALSE**. For example:

```
SELECT column1 FROM t1 WHERE EXISTS (SELECT * FROM t2);
```

Traditionally, an **EXISTS** subquery starts with **SELECT ***, but it could begin with **SELECT 5** or **SELECT column1** or anything at all. MySQL ignores the **SELECT** list in such a subquery, so it makes no difference.

For the preceding example, if **t2** contains any rows, even rows with nothing but **NULL** values, the **EXISTS** condition is **TRUE**. This is actually an unlikely example because a **[NOT] EXISTS** subquery almost always contains correlations. Here are some more realistic examples:

- What kind of store is present in one or more cities?

```
SELECT DISTINCT store_type FROM stores
  WHERE EXISTS (SELECT * FROM cities_stores
                WHERE cities_stores.store_type = stores.store_type);
```

- What kind of store is present in no cities?

```
SELECT DISTINCT store_type FROM stores
  WHERE NOT EXISTS (SELECT * FROM cities_stores
                    WHERE cities_stores.store_type =
stores.store_type);
```

- What kind of store is present in all cities?

```
SELECT DISTINCT store_type FROM stores s1
  WHERE NOT EXISTS (
    SELECT * FROM cities WHERE NOT EXISTS (
      SELECT * FROM cities_stores
       WHERE cities_stores.city = cities.city
       AND cities_stores.store_type = stores.store_type));
```

The last example is a double-nested **NOT EXISTS** query. That is, it has a **NOT EXISTS** clause within a **NOT EXISTS** clause. Formally, it answers the question "does a city exist with a store that is not in **Stores**"? But it is easier to say that a nested **NOT EXISTS** answers the question "is *x* **TRUE** for all *y*?"