# Lab 2. SQL Data definition and Integrity Constraints

## Entity Integrity

The primary key of a table must contain a unique, non-null value for each row. For example, each row of the `PropertyForRent` table has a unique value for the property number `propertyNo`, which uniquely identifies the property represented by that row:

**PRIMARY KEY**(propertyNo)

To define a composite primary key, we specify multiple column names in the PRIMARY KEY clause, separating each by a comma:

**PRIMARY KEY**(clientNo, propertyNo)

The PRIMARY KEY clause can be specified only once per table. However, it is still possible to ensure uniqueness for any alternate keys in the table using the keyword **UNIQUE**. Every column that appears in a UNIQUE clause must also be declared as NOT NULL:

clientNo **VARCHAR**(5) **NOT NULL**,
propertyNo **VARCHAR**(5) **NOT NULL**,
**UNIQUE** (clientNo, propertyNo)

## Referential Integrity

A foreign key is a column, or set of columns, that links each row in the child table containing the foreign key to the row of the parent table containing the matching candidate key value. Referential integrity means that, if the foreign key contains a value, that value must refer to an existing, valid row in the parent table. For example, to define the foreign key `branchNo` of the `PropertyForRent` table, we include the clause:

**FOREIGN KEY**(branchNo) **REFERENCES** Branch

SQL rejects any INSERT or UPDATE operation that attempts to create a foreign key value in a child table without a matching candidate key value in the parent table. The action SQL takes for any UPDATE or DELETE operation that attempts to update or delete a candidate key value in the parent table that has some matching rows in the child table is dependent on the **referential action** specified using the ON UPDATE and ON DELETE subclauses of the FOREIGN KEY clause. When the user attempts to delete a row from a parent table, and there are one or more matching rows in the child table, **MySQL** supports five options regarding the action to be taken, listed here:

- CASCADE: Delete or update the row from the parent table, and automatically delete or update the matching rows in the child table. Both ON DELETE CASCADE and ON UPDATE CASCADE are supported. Between two tables, do not define several ON UPDATE CASCADE clauses that act on the same column in the parent table or in the child table.
- SET NULL: Delete or update the row from the parent table, and set the foreign key column or columns in the child table to NULL. Both ON DELETE SET NULL and ON UPDATE SET NULL

clauses are supported. If you specify a SET NULL action, make sure that you have not declared the columns in the child table as NOT NULL.

- RESTRICT: Rejects the delete or update operation for the parent table. Specifying RESTRICT (or NO ACTION) is the same as omitting the ON DELETE or ON UPDATE clause.
- NO ACTION: A keyword from standard SQL. In MySQL, equivalent to RESTRICT. The MySQL Server rejects the delete or update operation for the parent table if there is a related foreign key value in the referenced table. Some database systems have deferred checks, and NO ACTION is a deferred check. In MySQL, foreign key constraints are checked immediately, so NO ACTION is the same as RESTRICT.
- SET DEFAULT: This action is recognized by the MySQL parser, but both InnoDB and NDB reject table definitions containing ON DELETE SET DEFAULT or ON UPDATE SET DEFAULT clauses.

For example, in the `PropertyForRent` table, the staff number `staffNo` is a foreign key referencing the `Staff` table. We can specify a deletion rule such that, if a staff record is deleted from the `Staff` table, the values of the corresponding `staffNo` column in the `PropertyForRent` table are set to NULL:

**FOREIGN KEY** (staffNo) **REFERENCES** Staff **ON DELETE SET NULL**

Similarly, the owner number `ownerNo` in the `PropertyForRent` table is a foreign key referencing the `PrivateOwner` table. We can specify an update rule such that, if an owner number is updated in the `PrivateOwner` table, the corresponding column(s) in the `PropertyForRent` table are set to the new value:

**FOREIGN KEY** (ownerNo) **REFERENCES** PrivateOwner **ON UPDATE CASCADE**

## Exercise

1. For tables you created in DreamHome database, identify integrity constraints and apply them in MySQL by recreating the tables again, using CREATE statements.
2. Try inserting into the tables with values that violates entity integrity constraints (primary key) and check the results.
3. Populate the new tables, and check referential integrity constraints (foreign key) by
   a) Deleting/updating tuples in parent table
   b) Inserting/updating tuples in child table without a matching candidate key value in the parent table.
4. Use ALTER TABLE statement in MySQL to add a column and drop a column in the table you have created. (use "`help alter table`" in MySQL to check syntax)
5. Use UPDATE statement in MySQL:
   a) Increase all manager's salary by 10%
   b) Change Julie Lee's position to supervisor and a salary increase of 5%
   c) Give an increase on rent by 2% on all properties in Glasgow