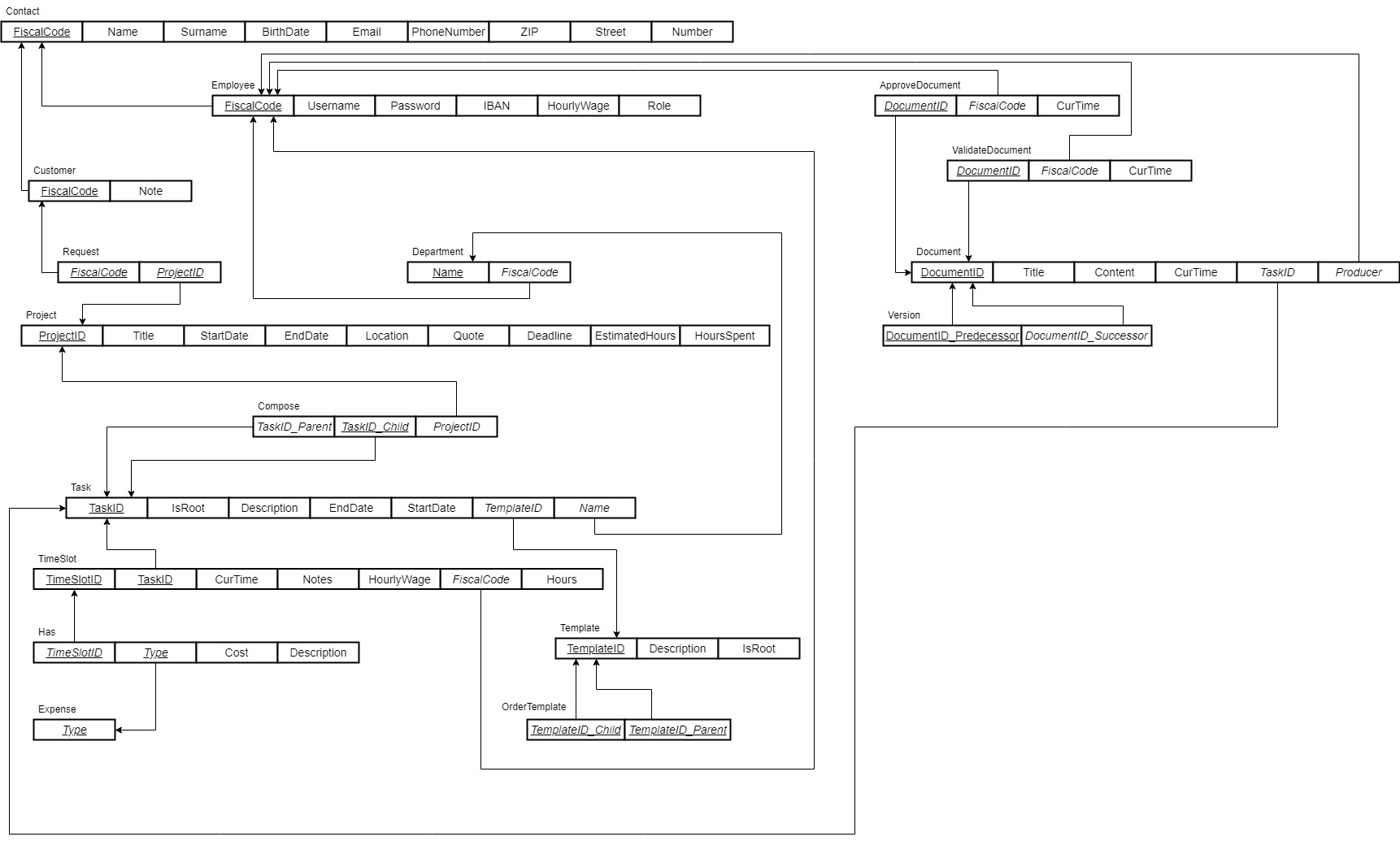
Variations to the Relational Schema

Figure 1 shows the relational schema. The type of the attribute ‘EstimatedHours’ in the relation Project has been changed from ‘Float’ to ‘Integer’ because the information does not need to be that precise.

*Figure 1: Relational Schema*

Physical Schema

In the following the SQL instructions to build the database in Figure 1 are reported. Note that the tables should be created in the correct order, as reported thereafter.

Trigger Function

The following trigger checks whether the HourlyWage of a certain Employee related to his dedicated TimeSlot is equal to the one stored in the table Employee.

-- When an Employee reports a certain Task, the HourlyWage declared in his dedicated Timeslot must be consistent with the one assigned to him in the Employee relation.

CREATE FUNCTION checkHourlyWage() RETURNS TRIGGER AS $$

BEGIN

-- Join the Employee table with the TimeSlot table

-- Check if the new HourlyWage inserted in TimeSlot is different from the one related to the associated Employee

IF NOT NEW.HourlyWage IN (SELECT E.HourlyWage

FROM Employee AS E INNER JOIN TimeSlot AS TS

ON E.FiscalCode = TS.FiscalCode

WHERE TS.TaskID = NEW.TaskID) THEN

-- If not, the new TimeSlot data cannot be inserted.

RAISE EXCEPTION 'Inconsistent Hourly Wage %.', NEW.HourlyWage USING HINT = 'Please check correctness';

END IF;

RETURN NEW;

END;

$$ LANGUAGE PLPGSQL;

CREATE TRIGGER CheckWage

AFTER INSERT ON TimeSlot

FOR EACH ROW

EXECUTE PROCEDURE checkHourlyWage();

The following trigger updates the total amount of HoursSpent for a certain Project whenever an Employee reports a Task specifying the number of Hours spent during a specific TimeSlot.

-- When an Employee inserts new data into the TimeSlot table, the total amout of HoursSpent for the Project must be updated by adding the new ones.

CREATE FUNCTION updateHoursSpent() RETURNS TRIGGER AS $$

BEGIN

-- Update the Project table

UPDATE Project

-- Sum the total amount of hours with the inserted ones

SET HoursSpent = Project.HoursSpent + NEW.Hours

-- Join TimeSlot, Task, Compose and Project tables and select the correct project from the Project table.

WHERE ProjectID IN (SELECT P.ProjectID

FROM Task AS T INNER JOIN TimeSlot AS TS

ON T.TaskID = TS.TaskID

INNER JOIN Compose AS C

ON TS.TaskID = C.Child

INNER JOIN Project AS P

ON C.ProjectID = P.ProjectID

WHERE C.Child = NEW.taskid);

RETURN NEW;

END;

$$ LANGUAGE PLPGSQL;

CREATE TRIGGER UpdateHours

AFTER INSERT ON TimeSlot

FOR EACH ROW

EXECUTE PROCEDURE updateHoursSpent();

Populate the Database: Example

Principal Queries

In this section, we report three queries to navigate the database:

1. Retrieve all the project-related information;
2. ????

JDBC Implementations of the Principal Queries and Visualization

Hereafter, we report a java class which read the data from the database and print the results on screen.