**Database User Guide**

**Overview**

|  |  |
| --- | --- |
| **Database Functionalities** | **Number of Scenarios** |
| Triggers | 4 |
| Function | 1 |
| Package | 2 |
| Procedures | 5 |

**2 Trigger, each with 2 scenarios**

* **Contract\_completion\_date**: This trigger will check if the new records inserted to and records updated in the table “contract” are valid before insertion and update. Valid records will have est\_complete\_date within 1 year after signed\_date. If the record is valid, it will be successfully inserted or updated. A message will be printed out to inform users about the success of insert/update. And, for insertion, only new est\_complete\_date will be printed out. For update, both new record’s and old record’s est\_complete\_date and signed\_date will be printed out. If the record is not valid (i.e., est\_complete\_date is beyond 1 year of signed\_date), an exception will be raised and the error message will be printed out.
* **Scenario 1**: A new house is sold and a new contract is created. The contract is signed on 09/10/2021, and the estimated completion date is 09/09/2022, which is within one year of the signed date. The record will be successfully inserted into the contract table and a message will be printed out indicating the success. And, estimated completion date and signed date for the new record will be printed out.

REM Insert a house with no contract to test on the trigger

INSERT INTO house

VALUES ('H0011', 549500.00, NULL,500.00, NULL,1);

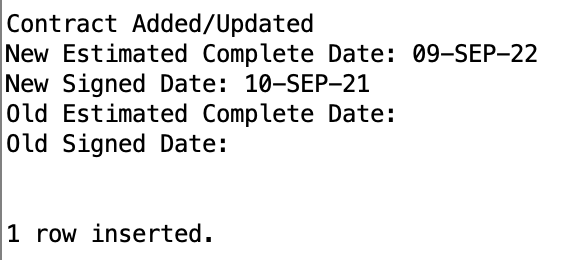
1 row inserted.

REM insert into contract with est\_completion\_date within required date

INSERT INTO contract

VALUES

('C0011', 'H0011', 'B0001', 'E0001', 'Portfolio Loan', 850000.00, 'EA001', 'BA003', '9-Sep-2022', '10-Sep-2021', 1, 1, 1);



* **Scenario 2**: A new house is sold and a new contract is created. The contract is signed on 09/10/2021, and the estimated completion date is 09/20/2022, which is beyond one year of the signed date. The record will be rejected and an exception will be raised. An error message will be printed out regarding the estimated completion date.

INSERT INTO constructionsheet

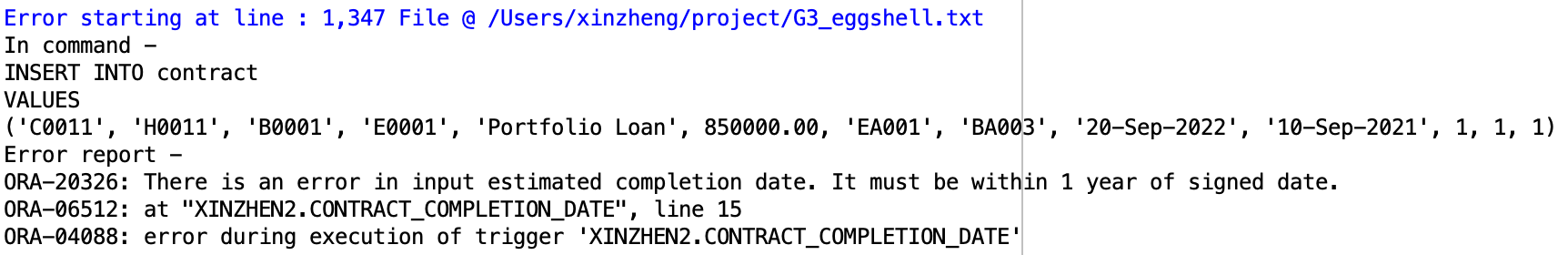
VALUES

('CS032', 2, '16-Jun-2022', '06-Aug-2022', 'E0010', 'H0010');

INSERT INTO constructionsheet

VALUES

('CS033', 2, '16-Jun-2022', '06-Aug-2022', 'E0010', 'H0010');



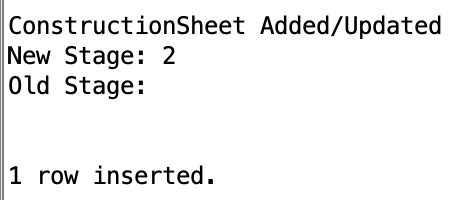
DELETE FROM constructionsheet where construction\_id = 'CS032';

* **Constructionsheet\_stage**: This trigger will check if the new record inserted to and the record updated in the table “constructionsheet” have a valid stage before insertion and update. The stage updated or inserted will need to be greater than the nearest stage by 1 according to the corresponding house number. For example, if the most recent stage of any task is 2, the newly inserted/updated stage must be 3. If the record is valid, a message informing the user about the success will be printed out. And, for insertion, the new stage will be printed out. For update, both the new stage and old stage will be printed out. If the record is invalid, an exception will be raised.
* **Scenario 1**: A new construction record is created in stage 2 of house H0010. Since it is greater than the most recent construction stage of house H0010 by 1, the role will be successfully inserted.

INSERT INTO constructionsheet

VALUES

('CS032', 2, '16-Jun-2022', '06-Aug-2022', 'E0010', 'H0010');

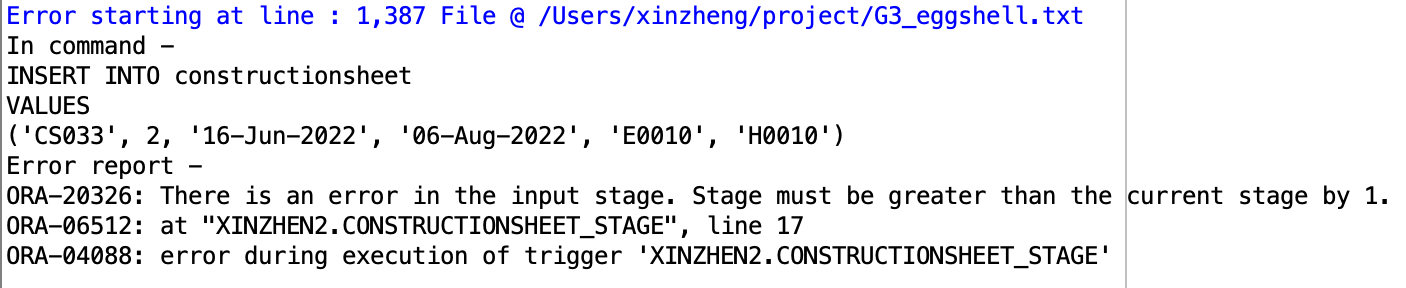


* **Scenario 2**: A new construction record is created for the house ‘H0010’. The stage of the record is 2. However, since we have entered a record of stage 2 for the house ‘H0010’, an error will be raised preventing the insertion. The new stage must be greater than the most recent stage of the house by at least 1. The error message will be printed out reminding the user about the incorrect stage.

INSERT INTO constructionsheet

VALUES

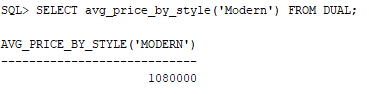
('CS033', 2, '16-Jun-2022', '06-Aug-2022', 'E0010', 'H0010');



DELETE FROM constructionsheet where construction\_id = 'CS032';

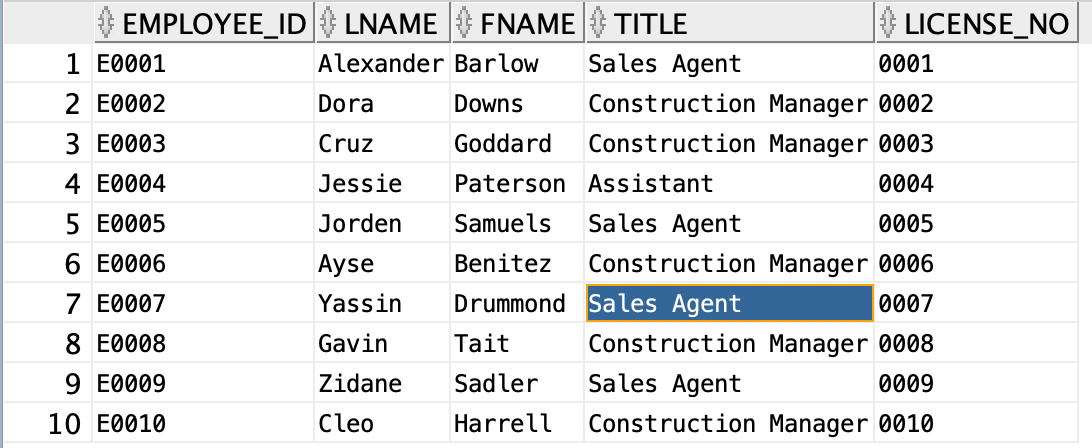
**1 Function with 1 scenario**

* **avg\_price\_by\_style**: a function that takes in style name as one parameter, calculates the average base price of all houses in the database of the given style, and returns the price.
* **Business Scenario:** a person is considering buying a house in the modern style, and he/she wants to know the average price outlook of houses in this style. The sales agent can then use this function to calculate the average base price, which turns out to be $1,080,000.



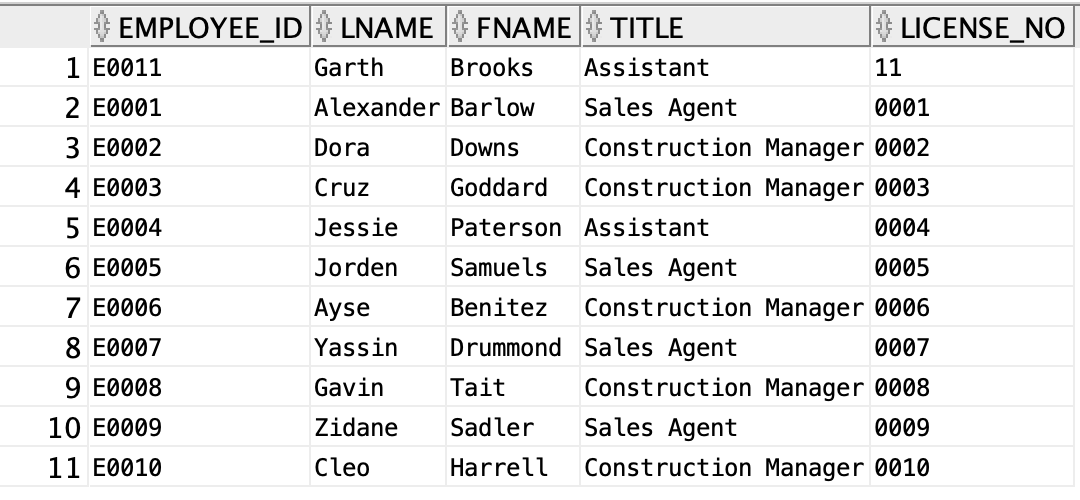
**1 Package with 2 scenarios**

* **Hiring and Firing employees**
  + By inserting a new employee’s Imployee\_id, firstname, lastname, position, licences, we can successfully add those information to employee table in our database
  + By inserting employee’ ID, the company can fire that person and delete related information in the database if the employee no longer serving any jobs
* **Scenario 1:** When the eggshell company wants to hire a new employee, the employee’s related information should be updated in its database.

The original database with employee table:

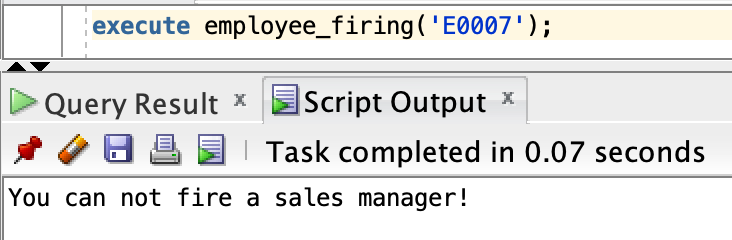
The updated database with employee table when add a new employee with employee id E0011, name Garth Brooks with title of assistant and its licence\_No is 0011.

execute employee\_hiring('E0011', 'Garth', 'Brooks', 'Assistant', '11');

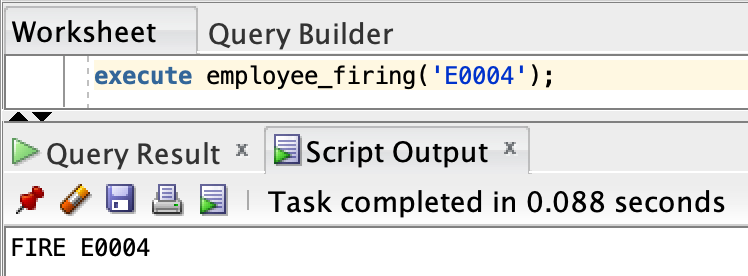
****

* **Scenario 2:** When the eggshell company wants to fire an employee, we can fire him or her by entering the employee’ ID. But be careful, the fired employee should not be a construction or sales manager for any house.

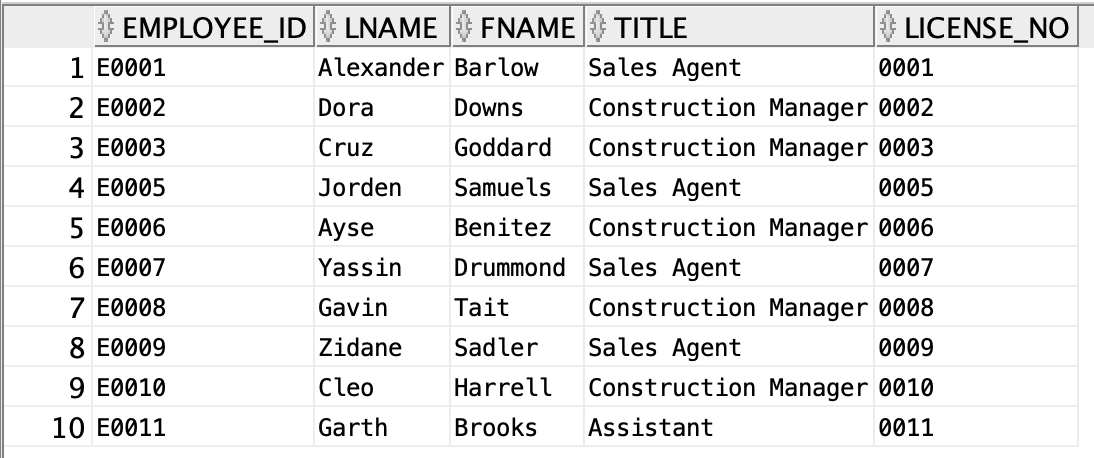
**1.Fire a sales manager (Not Applicable)**



**2.Fire an Assistant (Successful)**

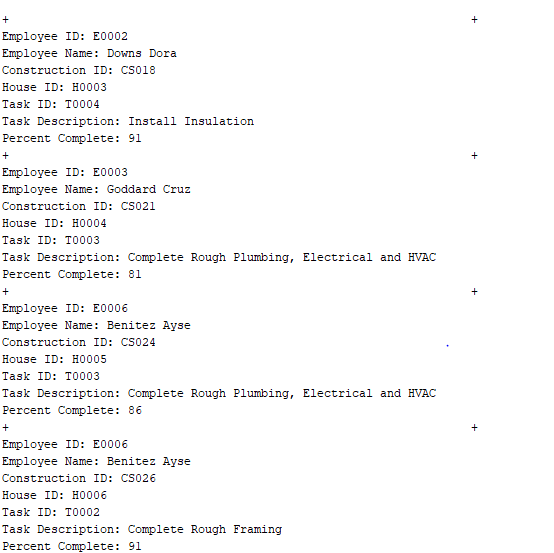


**3.Latest employees**



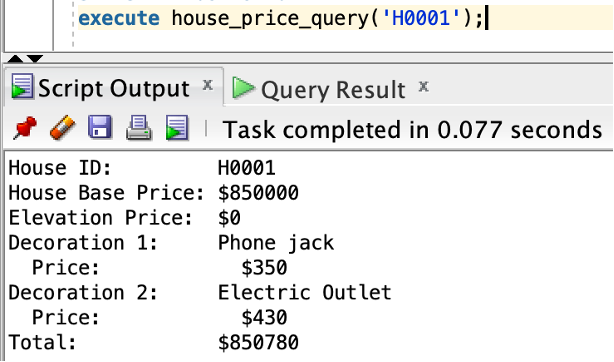
**2 Procedure, first with 1 scenario and second with 4 scenarios**

* **ongoing\_task:** a procedure that shows all construction managers who have ongoing constructions (percentage completed less than 100), their ongoing task, and the percentage completed.
* **Business Scenario:** When the Eggshell company want to track their ongoing construction tasks and find the responsible construction manager, they can use this procedure to list all incomplete tasks, the description of the tasks, the house it belongs to, and the information of construction managers. For example, they will find that the install insulation task in house H0003, which construction managed by Downs Dora, has finished 91%.

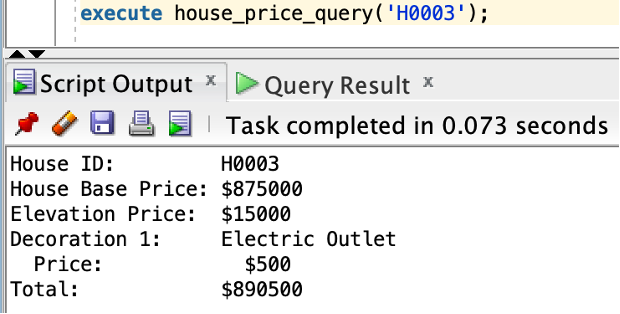


* **house\_price\_query:** The second procedure is used to ask the user to give a house\_id, and the system can show the house base price, elevation price, each decorator job and its price, and the total price of the chosen house. Each house has 1 base price, 1 elevation price but could have 0 or multiple decorator jobs and prices. For houses with multiple decorator jobs, the system will print out the price for each separately. And for houses with 0 decorator jobs so far, the system will print out “No decoration yet”. If the user types in a wrong house\_id that doesn’t exist, the system will give a warning message “No such house”.

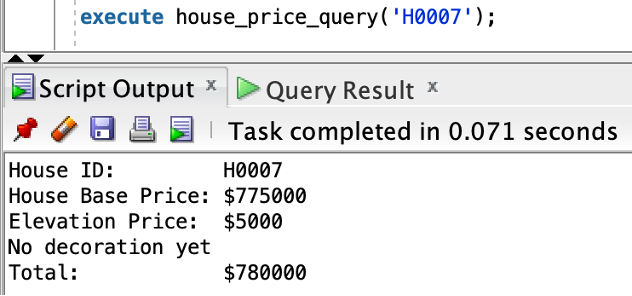
* **Scenario 1: multiple decoration jobs**



* **Scenario 2: Single decoration job**



* **Scenario 3: No decoration job yet**



* **Scenario 4: wrong input**

