**Report**

**Overview**

|  |  |
| --- | --- |
| **Content** | **Specifics** |
| Roles | sales\_representative  construction\_mgr  buyers |
| Scheduled Job | task\_schedule |
| Alternate Indexes | subdivision\_idx  room\_idx |
| De-normalization | Materialized view: construction\_progress |
| Test Cases | 2 scenarios |

**3 Roles**

* **Sales\_representative:** The role of sales representative will be able to see all the tables except for the table “employee”. This role will be able to select, insert, update and delete the tables “decoratorchoice”, “contract”, “house”, and “buyer”. A sales representative needs to create, modify or delete contracts and houses. The sales representative also needs to create, modify, or delete a buyer and create, modify or delete decorator choice sheets. This role will be able to select from “subdivision”, “"Subdivision-Style", “style”, “Option”, “Style-Elevation”, “schooldistrict”, “school”, “house”, “lot”, “elevation”, “room”, “constructionsheet”, “task”, “agent”, “bank”, and “employee”.

As the role **sales\_representative** has been created, during the actual test, it is possible that an error will be raised since the role has been created. Please change the name of the role if you would like to correctly create the role and grant it to a new user. In our run, the role is successfully created and the result is attached below along with the granting process to the new user and how to test it from the new user’s account.

REM create a role of sales representative

CREATE ROLE sales\_representative;

GRANT select, insert, update, delete ON decoratorchoice TO sales\_representative;

GRANT select, insert, update, delete ON contract TO sales\_representative;

GRANT select, insert, update, delete ON buyer TO sales\_representative;

GRANT SELECT, insert, update, delete ON house TO sales\_representative;

GRANT SELECT ON subdivision TO sales\_representative;

GRANT SELECT ON "Subdivision-Style" TO sales\_representative;

GRANT SELECT ON style TO sales\_representative;

GRANT SELECT ON "Option" TO sales\_representative;

GRANT SELECT ON "Style-Elevation" TO sales\_representative;

GRANT SELECT ON schooldistrict TO sales\_representative;

GRANT SELECT ON school TO sales\_representative;

GRANT SELECT ON lot TO sales\_representative;

GRANT SELECT ON elevation TO sales\_representative;

GRANT SELECT ON room TO sales\_representative;

GRANT SELECT ON constructionsheet TO sales\_representative;

GRANT SELECT ON task TO sales\_representative;

GRANT SELECT ON agent TO sales\_representative;

GRANT SELECT ON bank TO sales\_representative;

GRANT SELECT ON employee TO sales\_representative;

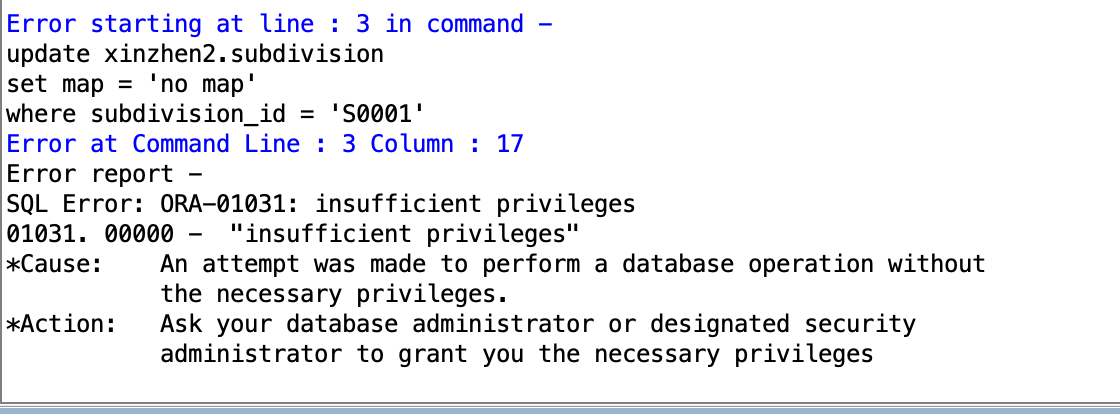
To test the role, we grant the role “sales\_representative” to the user advdb. To ensure the role has been successfully granted, open a new window and switch to the user advdb. Try to update the table “sudivision” on the other user’s schema. An error will be raised indicating that the user advdb does not have privilege to do so. Then, test to insert a new record into the table “house” on the other user’s scheme. The row will be successfully inserted.

GRANT sales\_representative TO advdb;

update SCHEMA\_USERNAME.subdivision

set map = 'no map'

where subdivision\_id = 'S0001';



INSERT INTO SCHEMA\_USERNAME.house

VALUES

('H0011', 789000, NULL, 800, NULL, 1);

1 row inserted.

DELETE FROM SCHEMA\_USERNAME.house

WHERE house\_id = 'H0011';

1 row deleted.

* **Construction\_Mgr:** The role of construction will be able to see all the tables except for the table “employee”. This role will be able to select, insert, update and delete the tables “constructionsheet”, “task”, “room”, and “house”, “elevation”, “subdivision”, “Subdivision-Style”, “style”, “Option”, and “Style-Elevation”. A construction manager needs to create, modify or delete houses. The construction manager also needs to create, modify, or delete any construction related table with the house. This role will be able to select from “lot”, “schooldistrict”, “school”, “agent”, “bank”, “decoratorchoice”, and “contract”.

As the role **Construction\_Mgr** has been created, during the actual test, it is possible that an error will be raised since the role has been created. Please change the name of the role if you would like to correctly create the role and grant it to a new user. In our run, the role is successfully created and the result is attached below along with the granting process to the new user and how to test it from the new user’s account.

REM create role of construction manager

CREATE ROLE construction\_mgr;

GRANT select, insert, update, delete ON constructionsheet TO construction\_mgr;

GRANT select, insert, update, delete ON task TO construction\_mgr;

GRANT select, insert, update, delete ON room TO construction\_mgr;

GRANT select, insert, update, delete ON house TO construction\_mgr;

GRANT select, insert, update, delete ON elevation TO construction\_mgr;

GRANT select, insert, update, delete ON subdivision TO construction\_mgr;

GRANT select, insert, update, delete ON "Subdivision-Style" TO construction\_mgr;

GRANT select, insert, update, delete ON style TO construction\_mgr;

GRANT select, insert, update, delete ON "Option" TO construction\_mgr;

GRANT select, insert, update, delete ON "Style-Elevation" TO construction\_mgr;

GRANT select ON lot TO construction\_mgr;

GRANT select ON schooldistrict TO construction\_mgr;

GRANT select ON school TO construction\_mgr;

GRANT select ON agent TO construction\_mgr;

GRANT select ON bank TO construction\_mgr;

GRANT select ON decoratorchoice TO construction\_mgr;

GRANT select ON contract TO construction\_mgr;

GRANT select ON buyer TO construction\_mgr;

To test the role, we grant the role “construction\_mgr” to the user advdb. To ensure the role has been successfully granted, open a new window and switch to the user advdb. Try to update the table “contract” on the other user’s schema. An error will be raised indicating that the user advdb does not have privilege to do so. Then, test to insert a new record into the table “style” on the other user’s scheme. The row will be successfully inserted.

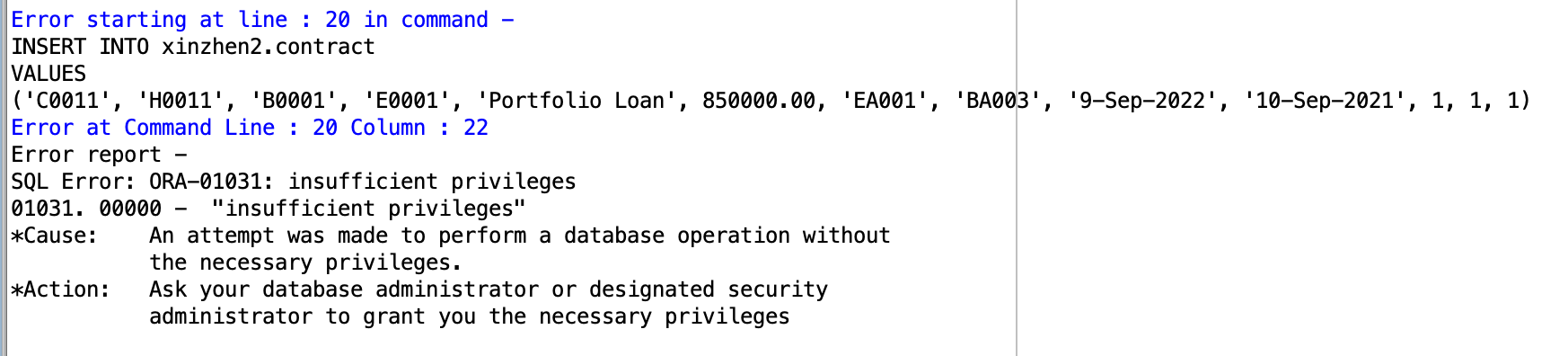
GRANT construction\_mgr TO advdb;

REM Insertion with no priviledge

INSERT INTO SCHEMA\_USERNAME.contract

VALUES

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



REM Insertion with priviledge

INSERT INTO SCHEMA\_USERNAME.style

VALUES

('new style', 4);

1 row inserted.

* **Buyers:** The role of buyers will be able to see the tables related to house. They will only be able to see construction stage 1, 4, and 7. Two views are created for users to see the construction progress. The buyers will be able to see tables “subdivision”, “Subdivision-Style”, “ style”, “Style-Elevation”, “schooldistrict”, “school”, “house”, “lot”, “elevation”, and “room”. The buyers will be able to see two views “construction\_stage” and “option\_view”.

As the role **Buyers** has been created, during the actual test, it is possible that an error will be raised since the role has been created. Please change the name of the role if you would like to correctly create the role and grant it to a new user. In our run, the role is successfully created and the result is attached below along with the granting process to the new user and how to test it from the new user’s account.

REM create role of buyer

CREATE ROLE buyers;

GRANT SELECT ON subdivision TO buyers;

GRANT SELECT ON "Subdivision-Style" TO buyers;

GRANT SELECT ON style TO buyers;

GRANT SELECT ON "Style-Elevation" TO buyers;

GRANT SELECT ON schooldistrict TO buyers;

GRANT SELECT ON school TO buyers;

GRANT SELECT ON house TO buyers;

GRANT SELECT ON lot TO buyers;

GRANT SELECT ON elevation TO buyers;

GRANT SELECT ON room TO buyers;

GRANT select ON construction\_stage to buyers;

GRANT SELECT ON option\_view to buyers;

To test the role, we grant the role “buyers” to the user advdb. To ensure the role has been successfully granted, open a new window and switch to the user advdb. Try to update the table “contract” on the other user’s schema. An error will be raised indicating that the user advdb does not have privilege to do so. Then, test to select a new record to see the view “construction\_stage” on the other user’s scheme.

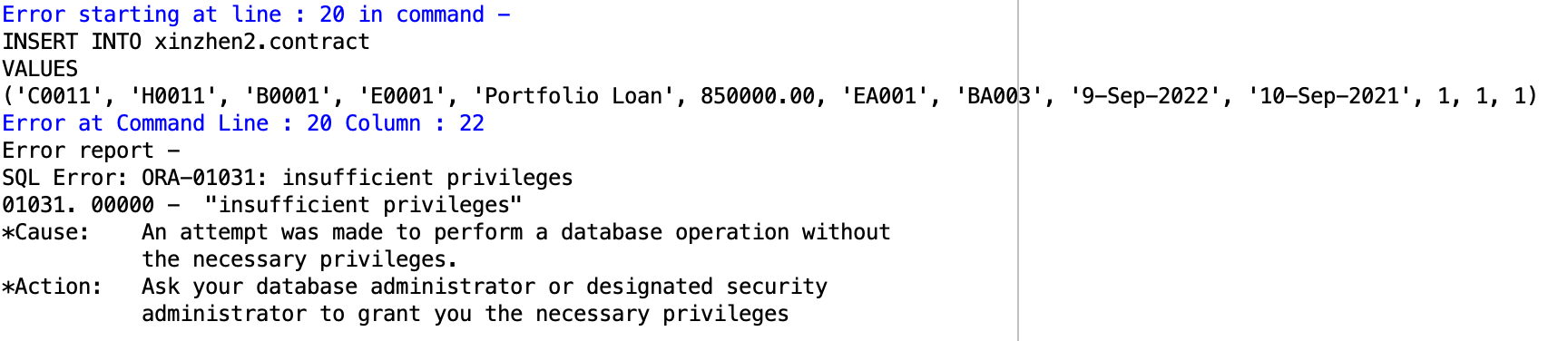
GRANT buyers TO advdb;

REM Insertion with no previledge

INSERT INTO SCHEMA\_USERNAME.contract

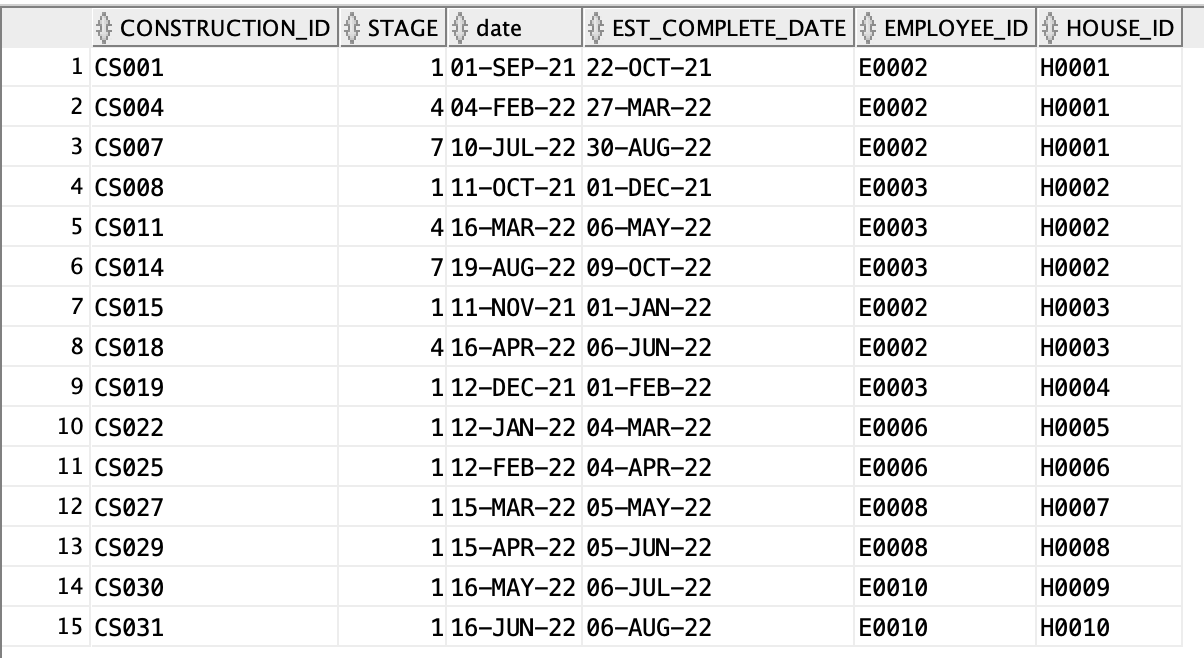
VALUES

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



REM Selection with previledge

SELECT \* FROM xinzhen2.construction\_stage;



**1 Scheduled Job**

* **task\_schedule:** A schedule job to update the column percent\_complete in tabletaskprogress. It is used to update the task progress on a daily basis. The scheduled job adds 1 to all percentage completed that are less than 100 every day starts from the job created date.

BEGIN

DBMS\_SCHEDULER.CREATE\_JOB (

job\_name => 'task\_schedule',

job\_type => 'PLSQL\_BLOCK',

job\_action => 'UPDATE taskprogress

SET percent\_complete = percent\_complete+1

WHERE percent\_complete <100;',

start\_date => sysdate,

repeat\_interval => 'FREQ=DAILY',

enabled => true);

END;

**2 Alternate Indexes**

* **subdivision\_idx**: unique alternate index for attribute subdivision\_name in table subdivision because subdivision\_name is a candidate key

CREATE INDEX subdivision\_idx ON subdivision (subdivision\_name);

This alternate index can help users retrieve information about subdivisions much more quickly without scanning through the whole table if they only know the subdivision names but not subdivision id, which is the primary key, thus making the whole process more time-saving and efficient.

Sample Query:

SELECT subdivision\_id,subdivision\_name, map

FROM subdivision

WHERE subdivision\_name = ‘Palm Springs’;

* **room\_idx**: unique alternate index for attribute room\_name in table room because room\_name is a candidate key

CREATE INDEX room\_idx ON room (room\_name);

This alternate index can help users retrieve information about room much more quickly without scanning through the whole table if they only know the room names but not the specific room id, which is the primary key, thus making the whole process more time-saving and efficient..

Sample Query:

SELECT hr.house\_id, r.room\_name, hr.floor,

hr."size", hr.num\_window

FROM room r JOIN "House-Room" hr USING(room\_id)

WHERE r.room\_name = 'Study';

**1 De-normalization**

* We denormalized the originally normalized entities ConstructionSheet, TaskProgress and Task by creating a materialized view **construction\_progress**. This view joined three tables to aggregate all tasks and task progress for constructions, which can lead to repeated/redundant information for construction\_id and task\_id but can also improve the read performance of our database, although at the expense of losing some write performance.
* In this case, when database users want to know the the tasks progress of one specific house, either in one specific stage or all stages, they don’t have to join all three tables mentioned above but directly retrieve the information from the materialized view construction\_progress, increasing query efficiency. This view will be updated whenever there’s an update in any of the three tables.

CREATE MATERIALIZED VIEW construction\_progress

REFRESH ON COMMIT

AS

SELECT c.house\_id, c.stage, t.task\_desc,tp.percent\_complete

FROM constructionsheet c, taskprogress tp, task t

WHERE c.construction\_id = tp.construction\_id

AND t.task\_id = tp.task\_id;

Before denormalization

SELECT c.house\_id, c.stage, t.task\_desc, tp.percent\_complete

FROM constructionsheet c JOIN taskprogress tp

USING(construction\_id)

JOIN task t USING(task\_id)

WHERE c.house\_id = 'H0003';

After denormalization

SELECT \*

FROM construction\_progress

WHERE house\_id = 'H0003';

**2 Test Cases for Business Use**

* **Scenario 1**

/\*

The first report demonstrates the basic information for all houses of Eggshell co.By running this query, for each house you can check the house id, city, construction manager, Sales manager, buyer, and its current status (complete or not. If not complete, which stage).

\*/

SELECT UNIQUE h.house\_id, l.city,

ecs.fname||' '||ecs.lname AS Construction\_Manager,

ec.fname||' '||ec.lname AS Sales\_Manager,

b.lname||' '||b.fname AS Buyer,

CASE -- status of houses, complete or in which stage

WHEN cs.stage = 7 AND tp.percent\_complete = 100

THEN 'Complete'

ELSE 'Stage '||cs.stage

END AS current\_progress

FROM house h, lot l, constructionsheet cs, contract c, employee ecs, employee ec, taskprogress tp, buyer b

WHERE h.house\_id = l.house\_id

AND h.house\_id = cs.house\_id

AND h.house\_id = c.house\_id

AND c.buyer\_id = b.buyer\_id

AND cs.employee\_id = ecs.employee\_id

AND c.employee\_id = ec.employee\_id

AND cs.construction\_id = tp.construction\_id

AND (h.house\_id, cs.stage) IN -- find the lastest stage number for each house

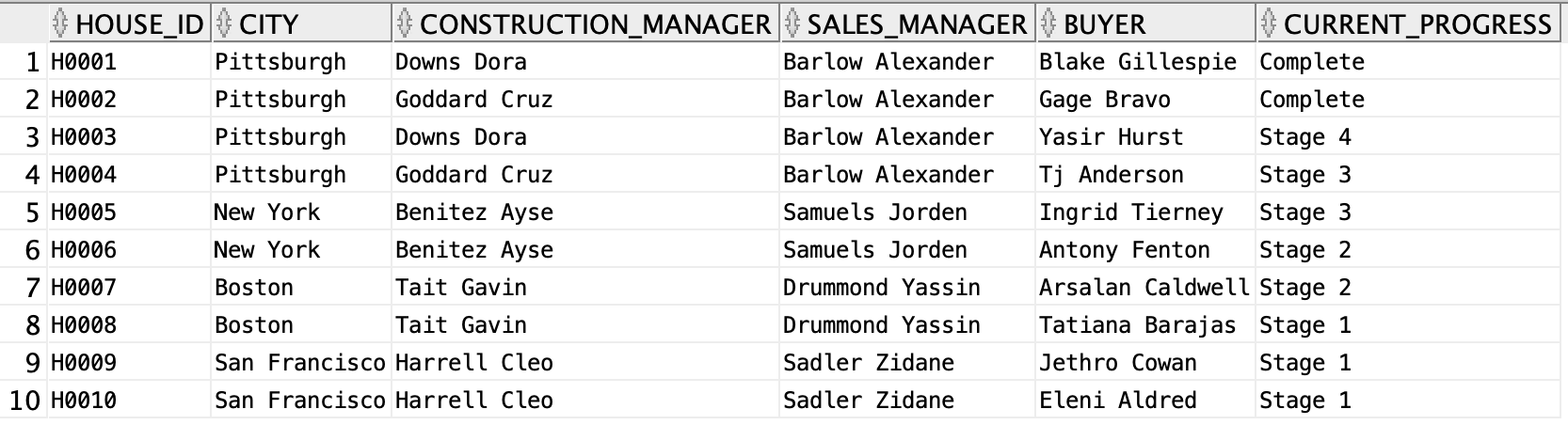
(SELECT h.house\_id, max(cs.stage)

FROM constructionsheet cs, house h

WHERE cs.house\_id = h.house\_id

GROUP BY h.house\_id)

ORDER BY house\_id;



* **Scenario 2**

/\*

The second report is about the details of houses,

Including the city, house base price, style, elevation type,Additional elevation price,

how many decoration jobs, total decoration cost, and the overall whole cost

\*/

SELECT h.house\_id, l.city, h.base\_price, s.style\_name, e.elevation\_name,

e.additional\_cost\_sketch AS elevation\_price, -- elevation price

count(dc.choice\_id) AS decoration#, -- how many decoration

NVL(sum(dc.item\_price),0) AS decoration\_price, -- decoration price

h.base\_price+e.additional\_cost\_sketch+NVL(sum(dc.item\_price),0) AS total\_price -- total price

FROM house h, lot l, "STYLE" s, elevation e, decoratorchoice dc

WHERE h.house\_id = l.house\_id

AND l.elevation\_name = e.elevation\_name

AND l.style\_name = s.style\_name

AND h.house\_id = dc.house\_id(+)

GROUP BY h.house\_id, l.city, h.base\_price, s.style\_name, e.elevation\_name, e.additional\_cost\_sketch

ORDER BY h.house\_id;

