Deakin University

Database Fundamentals

OnTrack Submission

Miniproject Part-2 - Database Implementation

Submitted By:
Sathiyanarayanan Senthil Kumar
s223789819
2023/09/12 22:01

Tutor:
Mehul Warade

Outcome	Weight
Structured Query Language (SQL)	♦♦♦♦♦

This task is to build my own database with the model I created in 4th week. In this task, I used various SQL commands to create, view and update data in my database. By completing this task, I got to know more about DML and DDL. So I believe this task is a very good example of the ULO - SQL.

September 12, 2023

Database Fundamentals Mini Project Part – 2

1. SQL Queries to create PROPERTY, ROOM, CUSTOMER, RESERVATION, REVIEW tables:

(A) PROPERTY:

```
CREATE TABLE PROPERTY(
PROPERTY_ID             INT             NOT NULL UNIQUE,
PROPERTY_NAME           VARCHAR(30)     NOT NULL,
PROPERTY_TYPE           VARCHAR(20)     NOT NULL,
PROPERTY_DESCRIPTION    VARCHAR(2000)       NOT NULL,
PROPERTY_PHONE          NUMERIC(10)     NOT NULL,
PROPERTY_STREET         VARCHAR(20)     NOT NULL,
PROPERTY_CITY           VARCHAR(15)     NOT NULL,
PROPERTY_COUNTRY        VARCHAR(15)     NOT NULL,
PRIMARY KEY(PROPERTY_ID));
```

(B) ROOM:

```
CREATE TABLE ROOM(
ROOM_NUM                INT             NOT NULL,
PROPERTY_ID             INT             NOT NULL,
ROOM_SPECIFICATION      VARCHAR(100)    NOT NULL,
ROOM_PRICE              NUMERIC(6,2)    NOT NULL,
ROOM_CAPACITY           INT             NOT NULL,
ROOM_STATUS             VARCHAR(15)     NOT NULL,
FOREIGN KEY (PROPERTY_ID) REFERENCES PROPERTY (PROPERTY_ID) ON DELETE CASCADE,
PRIMARY KEY(ROOM_NUM, PROPERTY_ID));
```

(C) CUSTOMER:

```
CREATE TABLE CUSTOMER(
CUST_ID                 INT             NOT NULL UNIQUE,
CUST_FNAME              VARCHAR(25)     NOT NULL,
CUST_LNAME              VARCHAR(25)     NOT NULL,
```

CUST_PHONE NUMERIC(10) NOT NULL,

CUST_EMAIL VARCHAR(100) NOT NULL,

CUST_CREDIT_CARD NUMERIC(20) NOT NULL,

CUST_CREDIT_EXP DATE NOT NULL,

PRIMARY KEY(CUST_ID));

(D) RESERVATION:

CREATE TABLE RESERVATION(

RESERVATION_ID INT NOT NULL UNIQUE,

PROPERTY_ID INT NOT NULL,

ROOM_NUM INT NOT NULL,

CUST_ID INT NOT NULL,

START_DATE DATE NOT NULL,

END_DATE DATE NOT NULL,

GUESTS INT NOT NULL,

RESERVATION_DATE DATE NOT NULL,

FOREIGN KEY (PROPERTY_ID) REFERENCES PROPERTY (PROPERTY_ID) ON DELETE CASCADE,

FOREIGN KEY (ROOM_NUM) REFERENCES ROOM (ROOM_NUM) ON DELETE CASCADE,

FOREIGN KEY (CUST_ID) REFERENCES CUSTOMER (CUST_ID) ON DELETE CASCADE,

PRIMARY KEY(RESERVATION_ID));

(E) REVIEW:

CREATE TABLE REVIEW(

REVIEW_ID INT NOT NULL UNIQUE,

PROPERTY_ID INT NOT NULL,

CUST_ID INT NOT NULL,

CUST_COMMENT VARCHAR(200) NOT NULL,

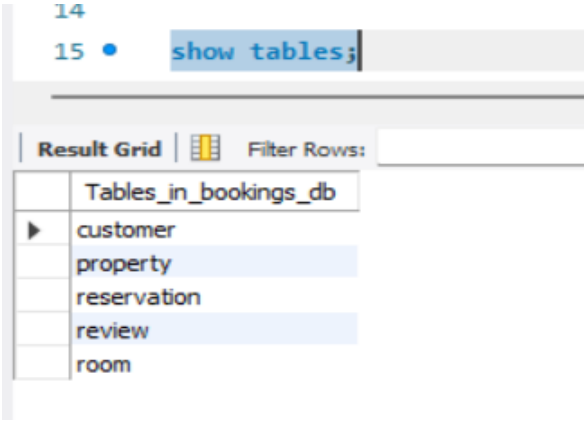
RATING_POINT INT NOT NULL,

REVIEW_TIMESTAMP DATETIME NOT NULL,

FOREIGN KEY (PROPERTY_ID) REFERENCES PROPERTY (PROPERTY_ID) ON DELETE CASCADE,

FOREIGN KEY (CUST_ID) REFERENCES CUSTOMER (CUST_ID) ON DELETE CASCADE,

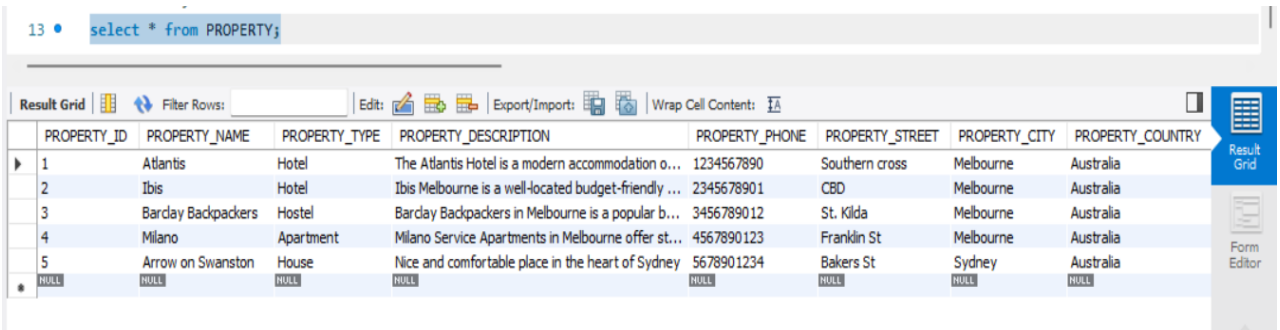
PRIMARY KEY(REVIEW_ID));



2. Few records in each table using INSERT Query:

(A) PROPERTY:

- *INSERT INTO PROPERTY VALUES(1, "Atlantis", "Hotel", "The Atlantis Hotel is a modern accommodation option located in the heart of Melbourne, Australia. It offers comfortable rooms, convenient access to Melbourne's attractions, and a range of amenities for travelers seeking a central stay.", 1234567890, "Southern cross", "Melbourne", "Australia");*
- *INSERT INTO PROPERTY VALUES(2, "Ibis", "Hotel", "Ibis Melbourne is a well-located budget-friendly hotel in the heart of Melbourne, offering comfortable accommodations and easy access to the city's attractions.", 2345678901, "CBD", "Melbourne", "Australia");*
- *INSERT INTO PROPERTY VALUES(3, "Barclay Backpackers", "Hostel", "Barclay Backpackers in Melbourne is a popular budget-friendly hostel, known for its central location and vibrant atmosphere, making it a favorite among backpackers and budget travelers.", 3456789012, "St. Kilda", "Melbourne", "Australia");*
- *INSERT INTO PROPERTY VALUES(4, "Milano", "Apartment", "Milano Service Apartments in Melbourne offer stylish and contemporary accommodation options with convenient access to the city's cultural and culinary attractions.", 4567890123, "Franklin St", "Melbourne", "Australia");*
- *INSERT INTO PROPERTY VALUES(5, "Arrow on Swanston", "House", "Nice and comfortable place in the heart of Sydney", 5678901234, "Bakers St", "Sydney", "Australia");*



(B) ROOM:

- *INSERT INTO ROOM VALUES(1, 1, "AC, TV, Fridge", 150, 2, "Available");*
- *INSERT INTO ROOM VALUES(2, 1, "Non-AC, TV", 100, 2, "Available");*
- *INSERT INTO ROOM VALUES(1, 2, "AC, TV, Fridge", 180, 2, "Available");*
- *INSERT INTO ROOM VALUES(2, 2, "AC, TV, Fridge, WI_FI", 200, 2, "Available");*
- *INSERT INTO ROOM VALUES(1, 3, "AC, TV", 50, 1, "Occupied");*
- *INSERT INTO ROOM VALUES(2, 3, "Non-AC, TV", 35, 1, "Available");*
- *INSERT INTO ROOM VALUES(1, 4, "AC, TV, Fridge, WI-FI", 250, 3, "Available");*
- *INSERT INTO ROOM VALUES(2, 4, "Non-AC, TV, Wi-FI", 200, 3, "Maintenance");*
- *INSERT INTO ROOM VALUES(1, 5, "AC, TV, Fridge, Wi-FI", 300, 6, "Available");*
- *INSERT INTO ROOM VALUES(2, 5, "Non-AC, TV, Fridge, Wi-FI", 350, 8, "Available");*

```
13 •  select * from ROOM;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

ROOM_NUM	PROPERTY_ID	ROOM_SPECIFICATION	ROOM_PRICE	ROOM_CAPACITY	ROOM_STATUS
1	1	AC, TV, Fridge	150.00	2	Available
1	2	AC, TV, Fridge	180.00	2	Available
1	3	AC, TV	50.00	1	Occupied
1	4	AC, TV, Fridge, WI-FI	250.00	3	Available
1	5	AC, TV, Fridge, Wi-FI	300.00	6	Available
2	1	Non-AC, TV	100.00	2	Available
2	2	AC, TV, Fridge, WI_FI	200.00	2	Available
2	3	Non-AC, TV	35.00	1	Available
2	4	Non-AC, TV, Wi-FI	200.00	3	Maintenance
2	5	Non-AC, TV, Fridge, Wi-FI	350.00	8	Available
NULL	NULL	NULL	NULL	NULL	NULL

(C) CUSTOMER:

- *INSERT INTO CUSTOMER VALUES (1, "Sathiyanarayanan", "Senthil Kumar", 4567894567, "sathiya@email.com", 1234567890987654, "2026-09-14");*
- *INSERT INTO CUSTOMER VALUES (2, "Sam", "Andrew", 9456745678, "sam@hotmail.com", 0987612954345678, "2025-08-21");*
- *INSERT INTO CUSTOMER VALUES (3, "Kim", "Mathers", 4556767894, "kim@yahoo.com", 1890982345677654, "2027-11-10");*
- *INSERT INTO CUSTOMER VALUES (4, "Mike", "Ross", 7894567456, "mike@gmail.com", 9098765412345678, "2023-12-30");*
- *INSERT INTO CUSTOMER VALUES (5, "Rachel", "Zane", 4589456767, "rachel@gmail.com", 8761230945678954, "2024-01-05");*

```
16 • select * from CUSTOMER;
17
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

CUST_ID	CUST_FNAME	CUST_LNAME	CUST_PHONE	CUST_EMAIL	CUST_CREDIT_CARD	CUST_CREDIT_EXP
1	Sathiyanarayanan	Senthil Kumar	4567894567	sathiya@email.com	1234567890987654	2026-09-14
2	Sam	Andrew	9456745678	sam@hotmail.com	987612954345678	2025-08-21
3	Kim	Mathers	4556767894	kim@yahoo.com	1890982345677654	2027-11-10
4	Mike	Ross	7894567456	mike@gmail.com	9098765412345678	2023-12-30
5	Rachel	Zane	4589456767	rachel@gmail.com	8761230945678954	2024-01-05
NULL	NULL	NULL	NULL	NULL	NULL	NULL

(D) RESERVATION:

- *INSERT INTO RESERVATION VALUES (1, 2, 1, 3, "2023-09-13", "2023-09-15", 2, "2023-09-11");*
- *INSERT INTO RESERVATION VALUES (2, 1, 2, 5, "2023-09-15", "2023-09-16", 2, "2023-09-13");*
- *INSERT INTO RESERVATION VALUES (3, 5, 2, 2, "2023-09-14", "2023-09-18", 2, "2023-09-09");*
- *INSERT INTO RESERVATION VALUES (4, 4, 1, 3, "2023-09-25", "2023-09-27", 2, "2023-09-08");*
- *INSERT INTO RESERVATION VALUES (5, 3, 2, 1, "2023-09-19", "2023-09-20", 2, "2023-09-02");*

```
12 • select * from RESERVATION;
13
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

RESERVATION_ID	PROPERTY_ID	ROOM_NUM	CUST_ID	START_DATE	END_DATE	GUESTS	RESERVATION_DATE
1	2	1	3	2023-09-13	2023-09-15	2	2023-09-11
2	1	2	5	2023-09-15	2023-09-16	2	2023-09-13
3	5	2	2	2023-09-14	2023-09-18	2	2023-09-09
4	4	1	3	2023-09-25	2023-09-27	2	2023-09-08
5	3	2	1	2023-09-19	2023-09-20	2	2023-09-02
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

(E) REVIEW:

- *INSERT INTO REVIEW VALUES (1, 2, 3, "Very good place!", 8, '2023-09-15 14:30:00');*
- *INSERT INTO REVIEW VALUES (2, 2, 3, "Very good place!", 8, "2023-09-15 18:25:20");*
- *INSERT INTO REVIEW VALUES (3, 5, 2, "Average", 6, "2023-09-18 21:10:43");*
- *INSERT INTO REVIEW VALUES (4, 4, 3, "Good", 7, "2023-09-27 20:50:44");*
- *INSERT INTO REVIEW VALUES (5, 3, 1, "Really Good!", 9, "2023-09-15 04:10:51");*

```
15 •  select * from REVIEW;
16
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Cont

REVIEW_ID	PROPERTY_ID	CUST_ID	CUST_COMMENT	RATING_POINT	REVIEW_TIMESTAMP
1	2	3	Very good place!	8	2023-09-15 14:30:00
2	2	3	Very good place!	8	2023-09-15 18:25:20
3	5	2	Average	6	2023-09-18 21:10:43
4	4	3	Good	7	2023-09-27 20:50:44
5	3	1	Really Good!	9	2023-09-15 04:10:51
NULL	NULL	NULL	NULL	NULL	NULL

3. SQL statement to add one additional column (attribute) in any one of the existing tables with a default value:

(A) In my database, I would like to add an additional attribute **"IS_LOCAL"** to **"RESERVATION"** table. The datatype of the attribute is **"CHAR",** and the default value is **"N"**. I am introducing this attribute to check whether the customer is from the same locality. This will be used to give a 10% discount for local customers.

ALTER TABLE RESERVATION

ADD COLUMN IS_LOCAL CHAR DEFAULT 'N';

```
11
12 •  select * from RESERVATION;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

RESERVATION_ID	PROPERTY_ID	ROOM_NUM	CUST_ID	START_DATE	END_DATE	GUESTS	RESERVATION_DATE	IS_LOCAL
1	2	1	3	2023-09-13	2023-09-15	2	2023-09-11	N
2	1	2	5	2023-09-15	2023-09-16	2	2023-09-13	N
3	5	2	2	2023-09-14	2023-09-18	2	2023-09-09	N
4	4	1	3	2023-09-25	2023-09-27	2	2023-09-08	N
5	3	2	1	2023-09-19	2023-09-20	2	2023-09-02	N
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

4. SQL statement to update the default value of the newly added column to a different value for certain rows based on a condition using any other column.

(A) We found out that the customers with **CUST_IDs 1 and 3** are local customers. So, I am planning to change the **IS_LOCAL** attribute's values of customers with IDs 1 and 3 to **"Y"**

UPDATE RESERVATION

SET IS_LOCAL = 'Y' WHERE CUST_ID = 3 OR CUST_ID = 1;

```
13 • select * from RESERVATION;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

RESERVATION_ID	PROPERTY_ID	ROOM_NUM	CUST_ID	START_DATE	END_DATE	GUESTS	RESERVATION_DATE	IS_LOCAL
1	2	1	3	2023-09-13	2023-09-15	2	2023-09-11	Y
2	1	2	5	2023-09-15	2023-09-16	2	2023-09-13	N
3	5	2	2	2023-09-14	2023-09-18	2	2023-09-09	N
4	4	1	3	2023-09-25	2023-09-27	2	2023-09-08	Y
5	3	2	1	2023-09-19	2023-09-20	2	2023-09-02	Y
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

5. SQL queries to interact with the database:

(A) SQL query to demonstrate the use of SELECT with INNER JOIN and ORDER BY:

Now, I wanted to see the property names (PROPERTY_NAME) of the properties from PROPERTY table ordered in ascending order of the START_DATE in RESERVATION table with attributes RESERVATION_ID, PROPERTY_ID, CUST_ID, RESERVATION_DATE, START_DATE

select R.RESERVATION_ID, R.PROPERTY_ID, P.PROPERTY_NAME, R.CUST_ID, R.RESERVATION_DATE, R.START_DATE from RESERVATION R inner join PROPERTY P where R.PROPERTY_ID = P.PROPERTY_ID ORDER BY R.START_DATE ASC;

```
12
13 • select R.RESERVATION_ID, R.PROPERTY_ID, P.PROPERTY_NAME, R.CUST_ID, R.RESERVATION
14
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

RESERVATION_ID	PROPERTY_ID	PROPERTY_NAME	CUST_ID	RESERVATION_DATE	START_DATE
1	2	Ibis	3	2023-09-11	2023-09-13
3	5	Arrow on Swanston	2	2023-09-09	2023-09-14
2	1	Atlantis	5	2023-09-13	2023-09-15
5	3	Barclay Backpackers	1	2023-09-02	2023-09-19
4	4	Milano	3	2023-09-08	2023-09-25

(B) SQL query to demonstrate the use of SELECT with WHERE and IN:

I wish to see the properties that are either a hotel or a house from PROPERTY table. The SQL query to implement that is,

*select * from PROPERTY where PROPERTY_TYPE IN ("Hotel", "House");*

```
13 • select * from PROPERTY where PROPERTY_TYPE IN ("Hotel", "House");
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

PROPERTY_ID	PROPERTY_NAME	PROPERTY_TYPE	PROPERTY_DESCRIPTION	PROPERTY_PHONE	PROPERTY_STREET	PROPERTY_CITY	PROPERTY_CO
1	Atlantis	Hotel	The Atlantis Hotel is a modern accommodation o...	1234567890	Southern cross	Melbourne	Australia
2	Ibis	Hotel	Ibis Melbourne is a well-located budget-friendly ...	2345678901	CBD	Melbourne	Australia
5	Arrow on Swanston	House	Nice and comfortable place in the heart of Sydney	5678901234	Bakers St	Sydney	Australia
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

(C) SQL query to demonstrate the use of at least one DATE function:

Now, I wish to see how many days each customer stays for a booking from the RESERVATION table. For this, I am subtracting END_DATE from START_DATE. The query to implement this is,

*select *, DATEDIFF(END_DATE, START_DATE) AS TOTAL_STAY_DAYS from RESERVATION;*

```
12
13 • select *, DATEDIFF(END_DATE, START_DATE) AS TOTAL_STAY_DAYS from RESERVATION;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

RESERVATION_ID	PROPERTY_ID	ROOM_NUM	CUST_ID	START_DATE	END_DATE	GUESTS	RESERVATION_DATE	IS_LOCAL	TOTAL_STAY_DAYS
1	2	1	3	2023-09-13	2023-09-15	2	2023-09-11	Y	2
2	1	2	5	2023-09-15	2023-09-16	2	2023-09-13	N	1
3	5	2	2	2023-09-14	2023-09-18	2	2023-09-09	N	4
4	4	1	3	2023-09-25	2023-09-27	2	2023-09-08	Y	2
5	3	2	1	2023-09-19	2023-09-20	2	2023-09-02	Y	1

(D) SQL statement to create a VIEW using a SELECT statement with a JOIN:

Here, I want to create a view called invoice which has the attributes CUSTOMER_FNAME, PROPERTY_NAME, ROOM_NUMBER, PRICE_PER_NIGHT, START_DATE, END_DATE, TOTAL_COST which will get information from PROPERTY table, ROOM table, CUSTOMER table and RESERVATION table.

```
CREATE VIEW INVOICE AS
SELECT
   C.CUST_FNAME,
   P.PROPERTY_NAME,
   RE.ROOM_NUM,
   R.ROOM_PRICE AS PRICE_PER_NIGHT,
   RE.START_DATE,
   RE.END_DATE,
   (DATEDIFF(RE.END_DATE, RE.START_DATE))*R.ROOM_PRICE AS TOTAL_COST
FROM
   RESERVATION RE
```

```
    JOIN
        CUSTOMER C ON RE.CUST_ID = C.CUST_ID
    JOIN
        ROOM R ON RE.ROOM_NUM = R.ROOM_NUM AND RE.PROPERTY_ID = R.PROPERTY_ID
    JOIN
        PROPERTY P ON RE.PROPERTY_ID = P.PROPERTY_ID;
```

```
27  select * from INVOICE;
28
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

CUST_FNAME	PROPERTY_NAME	ROOM_NUM	PRICE_PER_NIGHT	START_DATE	END_DATE	TOTAL_COST
Kim	Ibis	1	180.00	2023-09-13	2023-09-15	360.00
Rachel	Atlantis	2	100.00	2023-09-15	2023-09-16	100.00
Sam	Arrow on Swanston	2	350.00	2023-09-14	2023-09-18	1400.00
Kim	Milano	1	250.00	2023-09-25	2023-09-27	500.00
Sathiyanarayanan	Barclay Backpackers	2	35.00	2023-09-19	2023-09-20	35.00