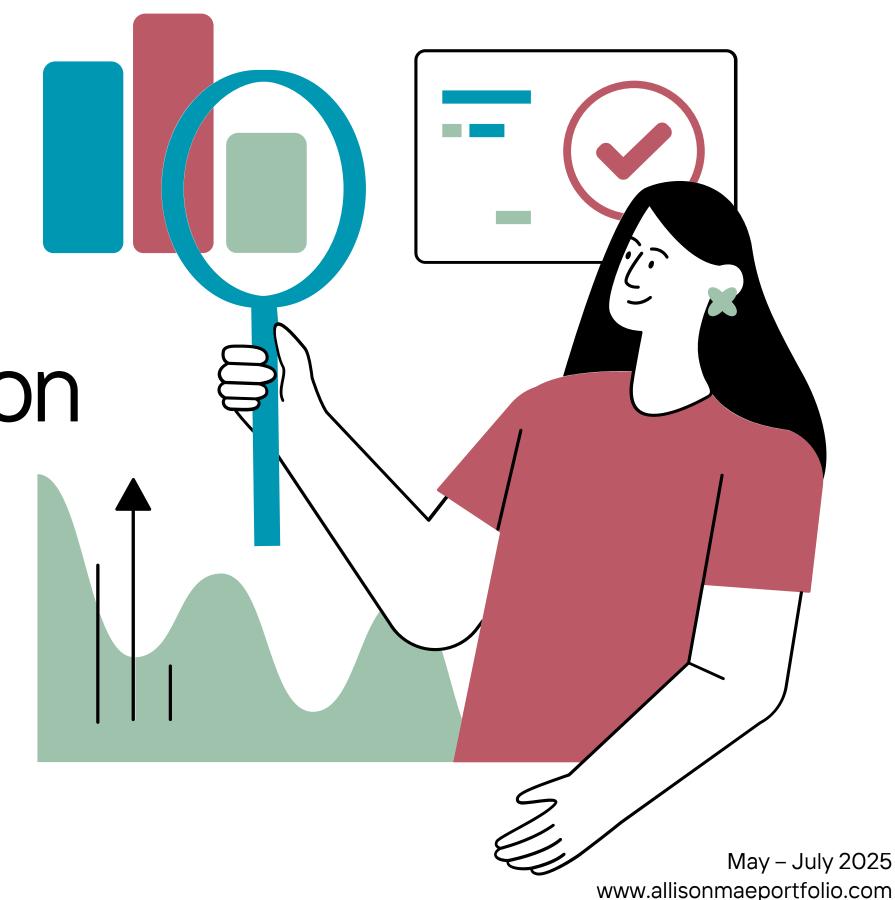
# StayWell

Student Accommodation

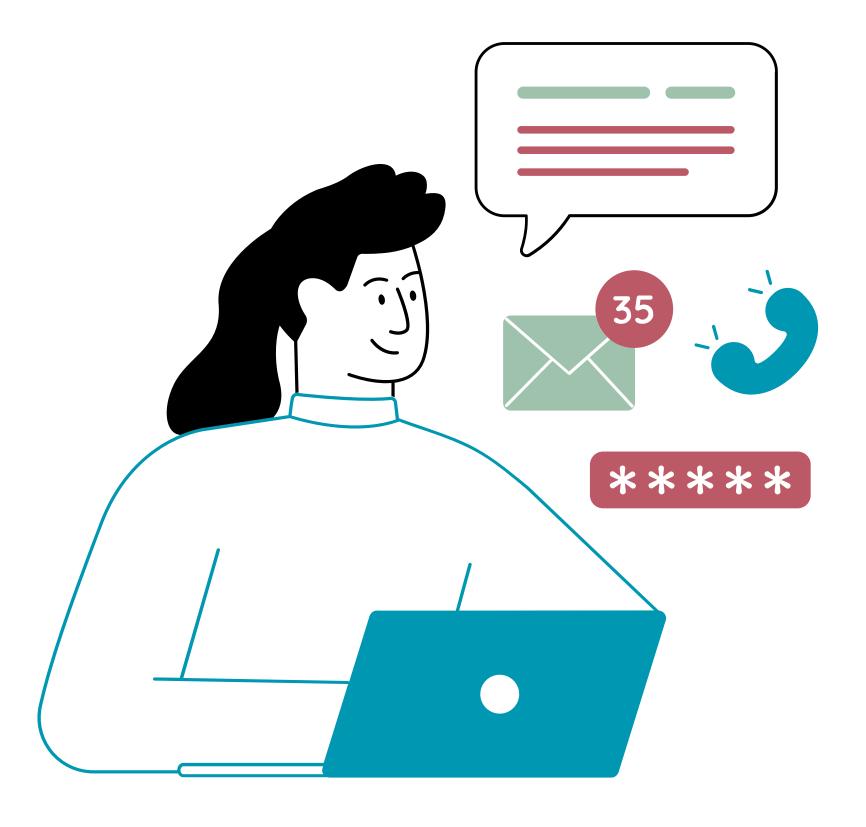
SQL PORTFOLIO PROJECT

- > DATABASE DESIGN
- > PROBLEM-SOLVING
- FOUNDATIONAL SQL



#### \* INTRODUCTION





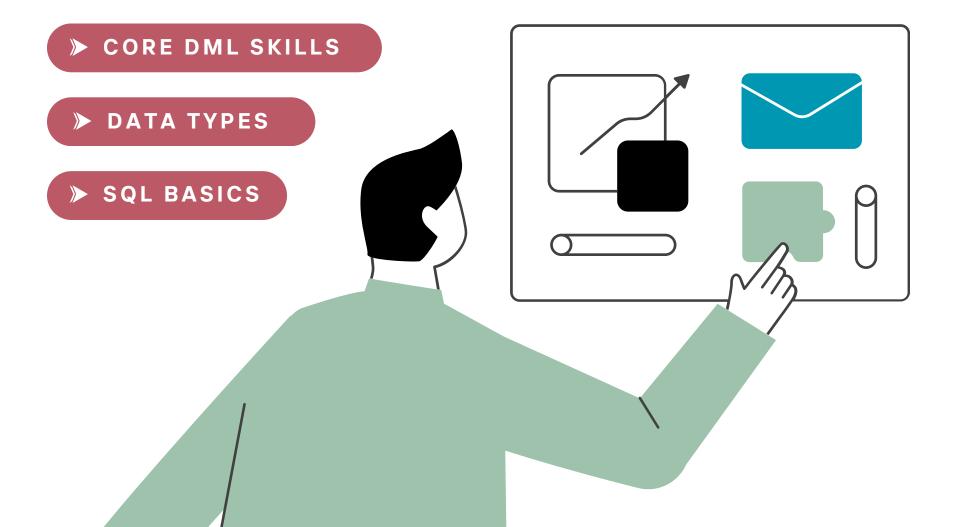
## Objective

This project demonstrates proficiency in SQL for relational database management, fulfilling the core objectives of the CTS 2433 course. The work centers on solving the business challenges of StayWell Student Accommodation, a property management company seeking to transition from manual to an ecommerce-based business model.

- How do we efficiently manage and retrieve information about all properties, owners, and service requests?
- How can we automate key business logic without manual calculations?
- How do we handle large datasets and ensure data integrity?
- How can we secure sensitive information and control user access?



## Database Creation & Foundational SQL



#### Objective

This section showcases foundational SQL skills by creating and managing the core tables for the StayWell Student Accommodation database. It includes table creation with specified data types, populating tables with initial data, and confirming table structure and data integrity using **DDL and DML** commands. The exercises demonstrate basic data definition and manipulation language proficiency.

#### **Exercise 1**

Create a table named SUMMER\_SCHOOL\_RENTALS – using CREATE TABLE command, defining columns and data types.

```
CREATE TABLE SUMMER_SCHOOL_RENTALS (
    PROPERTY_ID NUMERIC(2, 0) PRIMARY KEY,
    OFFICE_NUM NUMERIC(2, 0),
    ADDRESS CHAR(25),
    SQR_FT DECIMAL(5,0),
    BDRMS DECIMAL(2,0),
    FLOORS DECIMAL(2,0),
    WEEKLY_RENT DECIMAL(6, 2),
    OWNER_NUM CHAR(5)
);
```

Field	Туре	Null	Key	Default	Extra
PROPERTY_ID	decimal(2,0)	NO	PRI	NULL	
OFFICE_NUM	decimal(2,0)	YES		NULL	
ADDRESS	char(25)	YES		NULL	
SQR_FT	decimal(5,0)	YES		NULL	
BDRMS	decimal(2,0)	YES		NULL	
FLOORS	decimal(2,0)	YES		NULL	
WEEKLY_RENT	decimal(6,2)	YES		NULL	
OWNER_NUM	char(5)	YES		NULL	



Add a row to the SUMMER\_SCHOOL\_RENTALS table.

INSERT INTO SUMMER\_SCHOOL\_RENTALS
 VALUES (13, 1, '5867 Goodwin Ave', 1650, 2, 1,
 400.00, 'C0103');

#### Output:

PROPERTY_ID	OFFICE_NUM	ADDRESS	SQR_FT	BDRMS	FLOORS	WEEKLY_RENT	OWNER_NUM
13	1	5867 Goodwin Ave	1650	2	1	400	CO103

#### Exercise 3

Delete the SUMMER\_SCHOOL\_RENTALS table.

DROP TABLE SUMMER\_SCHOOL\_RENTALS;

#### **Exercise 4**

Run the provided script to create the StayWell database tables.



File available for download on GitHub (@allison-mae) and my portfolio website <a href="https://www.allisonmaeportfolio.com">www.allisonmaeportfolio.com</a>

#### Exercise 5

Confirm that the tables were created correctly using DESCRIBE command to view the structure, data types, and constraints of tables.

#### DESCRIBE OFFICE;

#### **Output:**

Field	Туре	Null	Key	Default	Extra
OFFICE_NUM	tinyint	NO	PRI	NULL	
OFFICE_NAME	varchar(50)	NO		NULL	
ADDRESS	varchar(100)	NO		NULL	
AREA	varchar(50)	NO		NULL	
CITY	varchar(20)	NO		NULL	
STATE	char(2)	NO		NULL	
ZIP_CODE	char(5)	NO		NULL	

#### DESCRIBE OWNER;

Field	Туре	Null	Key	Default	Extra
OWNER_NUM	char(5)	NO	PRI	NULL	
LAST_NAME	varchar(20)	NO		NULL	
FIRST_NAME	varchar(20)	NO		NULL	
ADDRESS	varchar(100)	NO		NULL	
CITY	varchar(20)	NO		NULL	
STATE	char(2)	NO		NULL	
ZIP_CODE	char(5)	NO		NULL	



#### DESCRIBE PROPERTY;

#### Output:

Field	Туре	Null	Key	Default	Extra
PROPERTY_ID	smallint	NO	PRI	NULL	
OFFICE_NUM	tinyint	NO		NULL	
ADDRESS	varchar(100)	NO		NULL	
SQR_FT	smallint	NO		NULL	
BDRMS	tinyint	NO		NULL	
FLOORS	tinyint	NO		NULL	
MONTHLY_RENT	smallint	YES		NULL	
OWNER_NUM	char(5)	NO		NULL	

#### DESCRIBE SERVICE\_CATEGORY;

#### Output:

Field	Туре	Null	Key	Default	Extra
CATEGORY_NUM	tinyint	NO	PRI	NULL	
CATEGORY_DESCRIPTION	varchar(100)	NO		NULL	

#### DESCRIBE SERVICE\_REQUEST;

#### Output:

Field	Туре	Null	Key	Default	Extra
SERVICE_ID	smallint	NO	PRI	NULL	
PROPERTY_ID	smallint	NO		NULL	
CATEGORY_NUMBER	tinyint	NO		NULL	
OFFICE_ID	tinyint	NO		NULL	
DESCRIPTION	varchar(200)	NO		NULL	
STATUS	varchar(200)	NO		NULL	
EST_HOURS	tinyint	NO		NULL	
SPENT_HOURS	tinyint	NO		NULL	
NEXT_SERVICE_DATE	date	YES		NULL	

#### DESCRIBE RESIDENTS;

Field	Туре	Null	Key	Default	Extra
RESIDENT_ID	smallint	NO	PRI	NULL	
FIRST_NAME	varchar(20)	NO		NULL	
SURNAME	varchar(20)	NO		NULL	
PROPERTY_ID	smallint	NO		NULL	



Confirm that data was added correctly by viewing the tables using SELECT command with the wildcard (\*) to view all rows and columns.

#### SELECT \* FROM OFFICE;

#### Output:

OFFICE_NUM	OFFICE_NAME	ADDRESS	AREA	CITY	STATE	ZIP_CODE
1	StayWell-Colombia City	1135 N. Wells Avenue	Colombia City	Seattle	WA	98118
2	StayWell-Georgetown	986 S. Madison Rd	Georgetown	Seattle	WA	98108

#### SELECT \* FROM OWNER;

#### Output:

OWNER_NUM	LAST_NAME	FIRST_NAME	ADDRESS	CITY	STATE	ZIP_CODE
AK102	Aksoy	Ceyda	411 Griffin Rd.	Seattle	WA	98131
BI109	Bianchi	Nicole	7990 Willow Dr.	New York	NY	10005
BU106	Burke	Ernest	613 Old Pleasant St.	Twin Falls	ID	83303
CO103	Cole	Meerab	9486 Circle Ave.	Olympia	WA	98506
JO110	Jones	Ammarah	730 Military Ave.	Seattle	WA	98126
KO104	Kowalczyk	Jakub	7431 S. Bishop St.	Bellingham	WA	98226
LO108	Lopez	Janine	9856 Pumpkin Hill Ln.	Everett	WA	98213
MO100	Moore	Elle-May	8006 W. Newport Ave.	Reno	NV	89508
PA101	Patel	Makesh	7337 Sheffield St.	Seattle	WA	98119
RE107	Redman	Seth	7681 Fordham St.	Seattle	WA	98119
SI105	Sims	Haydon	527 Primrose Rd.	Portland	OR	97203

#### SELECT \* FROM PROPERTY;

#### Output:

PROPERTY_ID	OFFICE_NUM	ADDRESS	SQR_FT	BDRMS	<b>FLOORS</b>	MONTHLY_RENT	OWNER_NUM
1	1	30 West Thomas Rd.	1600	3	1	1400	BU106
2	1	782 Queen Ln.	2100	4	2	1900	AK102
3	1	9800 Sunbeam Ave.	1005	2	1	1200	BI109
4	1	105 North Illinois Rd.	1750	3	1	1650	KO104
5	1	887 Vine Rd.	1125	2	1	1160	SI105
6	1	8 Laurel Dr.	2125	4	2	2050	MO100
7	2	447 Goldfield St.	1675	3	2	1700	CO103
8	2	594 Leatherwood Dr.	2700	5	2	2750	KO104
9	2	504 Windsor Ave.	700	2	1	1050	PA101
10	2	891 Alton Dr.	1300	3	1	1600	LO108
11	2	9531 Sherwood Rd.	1075	2	1	1100	JO110
12	2	2 Bow Ridge Ave.	1400	3	2	1700	RE107

#### SELECT \* FROM SERVICE\_CATEGORY;

CATEGORY_NUM	CATEGORY_DESCRIPTION
1	Plumbing
2	Heating
3	Painting
4	Electrical Systems
5	Carpentry
6	Furniture replacement



#### SELECT \* FROM SERVICE\_REQUEST;

#### Output:

SERVICE_ID	PROPERTY_ID	CATEGORY_NUM	OFFICE_ID	DESCRIPTION	STATUS	EST_HOURS	SPENT_HOURS	NEXT_SERVICE_DATE
1	11	2	2	The second bedroom upstairs is not heatir	Problem has	2	1	11/1/19
2	1	4	1	A new strip light is needed for the kitchen.	Scheduled	1	0	10/2/19
3	6	5	1	The bathroom door does not close properly	Service rep	3	1	11/9/19
4	2	4	1	New outlet has been requested for the firs	Scheduled	1	0	10/2/19
5	8	3	2	New paint job requested for the common a	Open	10	0	NULL
6	4	1	1	Shower is dripping when not in use.	Problem cor	4	2	10/7/19
7	2	2	1	Heating unit in the entrance smells like it-	Service rep	1	0	10/9/19
8	9	1	2	Kitchen sink does not drain properly.	Problem cor	6	2	11/12/19
9	12	6	2	New sofa requested.	Open	2	0	NULL

#### SELECT \* FROM RESIDENTS;

RESIDENT_ID	FIRST_NAME	SURNAME	PROPERTY_ID
1	Albie	O'Ryan	1
2	Tariq	Khan	1
3	Ismail	Salib	1
4	Callen	Beck	2
5	Milosz	Polansky	2
6	Ashanti	Lucas	2
7	Randy	Woodrue	2
8	Aislinn	Lawrence	3
9	Monique	French	3
10	Amara	Dejsuwan	4
12	Rosalie	Blackmore	4
13	Carina	Britton	4
14	Valentino	Ortega	5
15	Kaylem	Kent	5
16	Alessia	Wagner	6
17	Tyrone	Galvan	6
18	Constance	Fleming	6
19	Eamonn	Bain	6
20	Misbah	Yacob	7
21	Gianluca	Esposito	7
22	Elinor	Lake	7
23	Ray	Rosas	8
24	Damon	Caldwell	8
25	Dawood	Busby	8
26	Dora	Harris	8
27	Leroy	Stokes	8
28	Tamia	Hess	9
29	Amelia	Sanders	9
30	Zarah	Byers	10
31	Sara	Farrow	10
32	Delilah	Roy	10
33	Dougie	McDaniel	11
	Tahir	Halabi	11
	Mila	Zhikin	12
	Glenn	Donovan	12
	Zayn	Fowler	12



# Single-Table Queries

**▶** DATA RETRIEVAL

**▶** FILTERING & SORTING

**▶** DATA TRANSFORMATION

#### **Objective**

This section's objective is to build a foundational competency in data retrieval and manipulation from a single table. The exercises demonstrate the ability to apply various **conditions and logical operators** to filter data accurately. Additionally, it covers creating new, derived information through computed columns and structuring results in a specific order. The work also highlights skills in **summarizing data with aggregate functions** and organizing results into groups for deeper analysis.

#### Exercise 1

List the owner number, last name, and first name of every property owner.

SELECT OWNER\_NUM, LAST\_NAME, FIRST\_NAME
FROM OWNER;

LAST_NAME	FIRST NIANAE
LASI_INAME	FIRST_NAME
Aksoy	Ceyda
Bianchi	Nicole
Burke	Ernest
Cole	Meerab
Jones	Ammarah
Kowalczyk	Jakub
Lopez	Janine
Moore	Elle-May
Patel	Makesh
Redman	Seth
Sims	Haydon
	Aksoy Bianchi Burke Cole Jones Kowalczyk Lopez Moore Patel Redman





List the complete PROPERTY table (all rows and all columns).

SELECT \* FROM PROPERTY;

#### Output:

PROPERTY_ID	OFFICE_NUM	ADDRESS	SQR_FT	<b>BDRMS</b>	FLOORS	MONTHLY_RENT	OWNER_NUM
1	1	30 West Thomas Rd.	1600	3	1	1400	BU106
2	1	782 Queen Ln.	2100	4	2	1900	AK102
3	1	9800 Sunbeam Ave.	1005	2	1	1200	BI109
4	1	105 North Illinois Rd.	1750	3	1	1650	KO104
5	1	887 Vine Rd.	1125	2	1	1160	SI105
6	1	8 Laurel Dr.	2125	4	2	2050	MO100
7	2	447 Goldfield St.	1675	3	2	1700	CO103
8	2	594 Leatherwood Dr.	2700	5	2	2750	KO104
9	2	504 Windsor Ave.	700	2	1	1050	PA101
10	2	891 Alton Dr.	1300	3	1	1600	LO108
11	2	9531 Sherwood Rd.	1075	2	1	1100	JO110
12	2	2 Bow Ridge Ave.	1400	3	2	1700	RE107

#### Exercise 3

List the last name and first name of every owner who lives in Seattle.

SELECT LAST\_NAME, FIRST\_NAME
FROM OWNER WHERE CITY = 'Seattle';

#### Output:

LAST_NAME	FIRST_NAME
Aksoy	Ceyda
Jones	Ammarah
Patel	Makesh
Redman	Seth

#### **Exercise 4**

List the last name and first name of every owner who does not live in Seattle.

```
SELECT LAST_NAME, FIRST_NAME
FROM OWNER WHERE CITY <> 'Seattle';
```

#### **Output:**

LAST_NAME	FIRST_NAME
Bianchi	Nicole
Burke	Ernest
Cole	Meerab
Kowalczyk	Jakub
Lopez	Janine
Moore	Elle-May
Sims	Haydon

#### **Exercise 5**

List the property ID and office number for every property whose square footage is equal to or less than 1,400 square feet.

```
SELECT PROPERTY_ID, OFFICE_NUM
    FROM PROPERTY WHERE SQR_FT <= 1400;</pre>
```

PROPERTY_ID	OFFICE_NUM
3	1
5	1
9	2
10	2
11	2
12	2



List the office number and address for every property with three bedrooms.

```
SELECT OFFICE_NUM, ADDRESS
     FROM PROPERTY WHERE BDRMS = 3;
```

#### Output:

OFFICE_NUM	ADDRESS
1	30 West Thomas Rd.
1	105 North Illinois Rd.
2	447 Goldfield St.
2	891 Alton Dr.
2	2 Bow Ridge Ave.

#### Exercise 7

List the property ID for every property with two bedrooms that is managed by StayWell-Georgetown.

```
SELECT PROPERTY_ID
FROM PROPERTY WHERE BDRMS = 2 AND OFFICE_NUM = 2;
```

#### Output:

PROPERTY_ID	
	9
	11

#### Exercise 8

List the property ID for every property with a monthly rent that is between \$1,350 and \$1,750.

```
SELECT PROPERTY_ID
FROM PROPERTY
WHERE MONTHLY_RENT BETWEEN 1350 AND 1750;
```

#### **Output:**

PROPERTY_ID	
	1
	4
	7
	10
	12

#### **Exercise 9**

List the property ID for every property with a monthly rent that is between \$1,350 and \$1,750.

```
SELECT PROPERTY_ID

FROM PROPERTY

WHERE OFFICE_NUM = 1 AND MONTHLY_RENT < 1500;
```

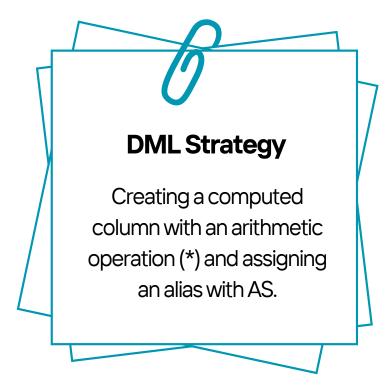
PROPERTY_ID	
	1
	3
	5



Labor is billed at the rate of \$35 per hour. List the property ID, category number, estimated hours, and estimated labor cost for every service request. To obtain the estimated labor cost, multiply the estimated hours by 35. Use the column name ESTIMATED\_COST for the estimated labor cost.

SELECT PROPERTY\_ID, CATEGORY\_NUMBER, EST\_HOURS, (EST\_HOURS \* 35) AS ESTIMATED\_COST
FROM SERVICE\_REQUEST;

PROPERTY_ID	CATEGORY_NUMBER	EST_HOURS	ESTIMATED_COST
11	2	2	70
1	4	1	35
6	5	3	105
2	4	1	35
8	3	10	350
4	1	4	140
2	2	1	35
9	1	6	210
12	6	2	70



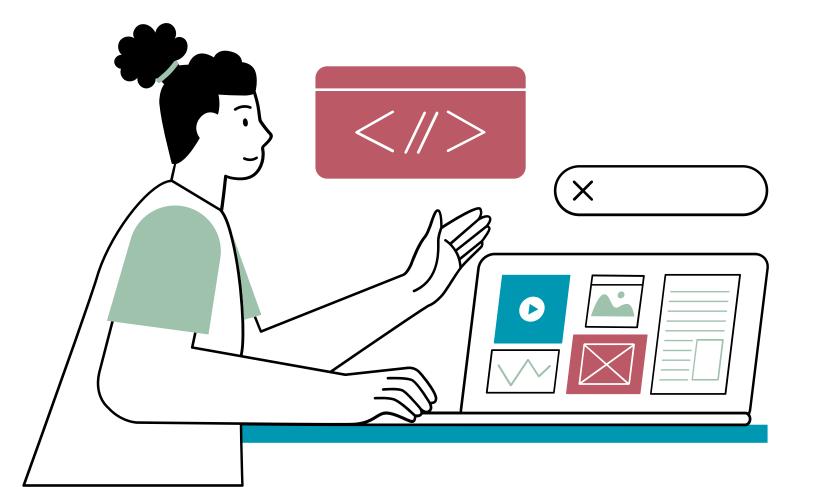


# Multiple-Table Queries

**>** JOINS

**▶** SUBQUERIES

**▶** SET OPERATIONS



#### Objective

This section focuses on retrieving and analyzing data from multiple related tables simultaneously. The exercises showcase proficiency in combining information using various **join techniques**, including self-joins and both correlated and non-correlated subqueries. The work demonstrates an ability to use **advanced operators** and **logical conditions** to perform complex data filtering, enabling comprehensive analysis of interconnected data across the database.

#### Exercise 1

For every property, list the management office number, address, monthly rent, owner number, owner's first name, and owner's last name.

SELECT P.OFFICE\_NUM, P.ADDRESS, P.MONTHLY\_RENT, P.OWNER\_NUM,
O.FIRST\_NAME, O.LAST\_NAME
FROM PROPERTY P, OWNER O
WHERE P.OWNER\_NUM = O.OWNER\_NUM;

OFFICE_NUM	ADDRESS	MONTHLY_RENT	OWNER_NUM	FIRST_NAME	LAST_NAME
1	30 West Thomas Rd.	1400	BU106	Ernest	Burke
1	782 Queen Ln.	1900	AK102	Ceyda	Aksoy
1	9800 Sunbeam Ave.	1200	BI109	Nicole	Bianchi
1	105 North Illinois Rd.	1650	KO104	Jakub	Kowalczyk
1	887 Vine Rd.	1160	SI105	Haydon	Sims
1	8 Laurel Dr.	2050	MO100	Elle-May	Moore
2	447 Goldfield St.	1700	CO103	Meerab	Cole
2	594 Leatherwood Dr.	2750	KO104	Jakub	Kowalczyk
2	504 Windsor Ave.	1050	PA101	Makesh	Patel
2	891 Alton Dr.	1600	LO108	Janine	Lopez
2	9531 Sherwood Rd.	1100	JO110	Ammarah	Jones
2	2 Bow Ridge Ave.	1700	RE107	Seth	Redman



For every completed or open service request, list the property ID, description, and status.

SELECT PROPERTY\_ID, DESCRIPTION, STATUS
FROM SERVICE\_REQUEST;

#### Output:

PROPERTY_ID	DESCRIPTION	STATUS
11	The second bedroom upstairs is not heating up at night.	Problem has been confirmed. central heating engineer has been scheduled.
1	A new strip light is needed for the kitchen.	Scheduled
6	The bathroom door does not close properly.	Service rep has confirmed issue. Scheduled to be refitted.
2	New outlet has been requested for the first upstairs bedroom.	Scheduled
8	New paint job requested for the common area (lounge).	Open
4	Shower is dripping when not in use.	Problem confirmed. Plumber has been scheduled.
2	Heating unit in the entrance smells like it's burning.	Service rep confirmed the issue to be dust in the heating unit. To be cleaned.
9	Kitchen sink does not drain properly.	Problem confirmed. Plumber scheduled.
12	New sofa requested.	Open

#### **Exercise 3**

For every service request for furniture replacement, list the property ID, management office number, address, estimated hours, spent hours, owner number, and owner's last name.

```
SELECT SR.PROPERTY_ID, SR.OFFICE_ID, P.ADDRESS,
SR.EST_HOURS, SR.SPENT_HOURS, P.OWNER_NUM, O.LAST_NAME
FROM SERVICE_REQUEST SR, PROPERTY P, OWNER O
WHERE SR.CATEGORY_NUMBER = 6
AND SR.PROPERTY_ID = P.PROPERTY_ID
AND P.OWNER_NUM = O.OWNER_NUM;
```

#### **Output:**

PROPERTY_ID	OFFICE_ID	ADDRESS	EST_HOURS	SPENT_HOURS	OWNER_NUM	LAST_NAME
12	2	2 Bow Ridge Ave.	2	0	RE107	Redman

#### **Exercise 4**

List the first and last names of all owners who own a two-bedroom property.

```
SELECT O.FIRST_NAME, O.LAST_NAME
FROM OWNER O
WHERE O.OWNER_NUM IN (
SELECT OWNER_NUM
FROM PROPERTY
WHERE BDRMS = 2
);
```

FIRST_NAME	LAST_NAME
Nicole	Bianchi
Haydon	Sims
Makesh	Patel
Ammarah	Jones



List the property IDs of any pair of properties that have the same number of bedrooms.

```
SELECT F.PROPERTY_ID AS Property1_ID, S.PROPERTY_ID
AS Property2_ID
FROM PROPERTY F, PROPERTY S
WHERE F.BDRMS = S.BDRMS
AND F.PROPERTY_ID < S.PROPERTY_ID
ORDER BY Property1_ID, Property2_ID;</pre>
```

#### Output:

Property1_ID	Property2_ID
1	4
1	7
1	10
1	12
2	6
3	5
3	9
3	11
4	7
4	10
4	12
5	9
5	11
7	10
7	12
9	11
10	12

#### Exercise 6

List the square footage, owner number, owner last name, and owner first name for each property managed by the Columbia City office.

```
SELECT P.SQR_FT, P.OWNER_NUM, O.LAST_NAME,
O.FIRST_NAME
FROM PROPERTY P, OWNER O
WHERE P.OWNER_NUM = O.OWNER_NUM
AND P.OFFICE_NUM = 1;
```

SQR_FT	OWNER_NUM	LAST_NAME	FIRST_NAME
1600	BU106	Burke	Ernest
2100	AK102	Aksoy	Ceyda
1005	BI109	Bianchi	Nicole
1750	KO104	Kowalczyk	Jakub
1125	SI105	Sims	Haydon
2125	MO100	Moore	Elle-May



List the office number, address, and monthly rent for properties whose owners live in Washington state or own two-bedroom properties.

```
SELECT OFFICE_NUM, ADDRESS, MONTHLY_RENT
    FROM PROPERTY
    WHERE BDRMS = 2
    OR OWNER_NUM IN (SELECT OWNER_NUM FROM OWNER
WHERE STATE = 'WA');
```

#### Output:

OFFICE_NUM	ADDRESS	MONTHLY_RENT
1	782 Queen Ln.	1900
1	9800 Sunbeam Ave.	1200
1	105 North Illinois Rd.	1650
1	887 Vine Rd.	1160
2	447 Goldfield St.	1700
2	594 Leatherwood Dr.	2750
2	504 Windsor Ave.	1050
2	891 Alton Dr.	1600
2	9531 Sherwood Rd.	1100
2	2 Bow Ridge Ave.	1700

#### **Exercise 8**

List the office number, address, and monthly rent for properties whose owners live in Washington state AND own a two-bedroom property.

```
SELECT OFFICE_NUM, ADDRESS, MONTHLY_RENT
    FROM PROPERTY
    WHERE BDRMS = 2
    AND OWNER_NUM IN (SELECT OWNER_NUM FROM OWNER
WHERE STATE = 'WA');
```

#### Output:

OFFICE_NUM	ADDRESS	MONTHLY_RENT	
2	504 Windsor Ave.	1050	
2	9531 Sherwood Rd.	1100	

#### Exercise 9

Find the service ID and property ID for each service request whose estimated hours are greater than the number of estimated hours of AT LEAST ONE service request on which the category number is 5.

```
SELECT SERVICE_ID, PROPERTY_ID
FROM SERVICE_REQUEST
WHERE EST_HOURS > ANY (
        SELECT EST_HOURS FROM SERVICE_REQUEST
        WHERE CATEGORY_NUMBER = 5
);
```

SERVICE_ID	PROPERTY_ID
5	8
6	4
8	9



List the address, square footage, owner number, service ID, number of estimated hours, and number of spent hours for each service request on which the category number is 4.

```
SELECT P.ADDRESS, P.SQR_FT, P.OWNER_NUM, SR.SERVICE_ID, SR.EST_HOURS, SR.SPENT_HOURS
FROM PROPERTY P, SERVICE_REQUEST SR
WHERE P.PROPERTY_ID = SR.PROPERTY_ID
AND SR.CATEGORY_NUMBER = 4;
```

#### **Output:**

ADDRESS	SQR_FT	OWNER_NUM	SERVICE_ID	EST_HOURS	SPENT_HOURS
30 West Thomas Rd.	1600	BU106	2	1	0
782 Queen Ln.	2100	AK102	4	1	0

# DML Strategy Joining two tables (PROPERTY and SERVICE\_REQUEST) and filtering the results based on a specific service category.



# Data Manipulation and Structure

**▶ DYNAMIC TABLES** 

**▶** ALTERING SCHEMA

**>> BULK DATA OPERATIONS** 

#### **Objective**

This section focuses on data manipulation and table structure management using advanced DML and DDL commands. It covers **creating new tables** from existing data, performing **conditional and bulk updates**, and adding, changing, and deleting data. The exercises also demonstrate **how to alter a table's schema** by adding new columns and modifying existing column properties, which are critical skills for database maintenance.

#### Exercise 1

Create a LARGE\_PROPERTY table.

```
CREATE TABLE LARGE_PROPERTY

(

OFFICE_NUM DECIMAL(2, 0),

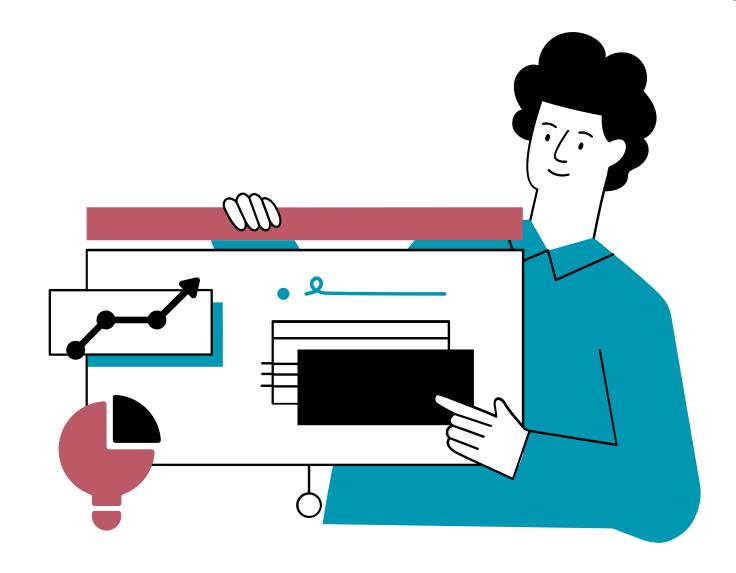
ADDRESS CHAR(25) PRIMARY KEY,

BDRMS DECIMAL(2, 0),

FLOORS DECIMAL(2, 0),

MONTHLY_RENT DECIMAL(6, 2),

OWNER_NUM CHAR(5)
);
```



Field	Туре	Null	Key	Default	Extra
OFFICE_NUM	decimal(2,0)	YES		NULL	
ADDRESS	char(25)	NO	PRI	NULL	
BDRMS	decimal(2,0)	YES		NULL	
FLOORS	decimal(2,0)	YES		NULL	
MONTHLY_RENT	decimal(6,2)	YES		NULL	
OWNER_NUM	char(5)	YES		NULL	



Insert into the LARGE\_PROPERTY table the office number, address, bedrooms, baths, monthly rent, and owner number for those properties whose square footage is greater than 1,500 square feet.

```
INSERT INTO LARGE_PROPERTY
    (SELECT OFFICE_NUM, ADDRESS, BDRMS, FLOORS,
MONTHLY_RENT, OWNER_NUM
    FROM PROPERTY
    WHERE (SQR_FT > 1500));
```

#### Output:

OFFICE_NUM	ADDRESS	BDRMS	FLOORS	MONTHLY_RENT	OWNER_NUM
1	105 North Illinois Rd.	3	1	1650	KO104
1	30 West Thomas Rd.	3	1	1400	BU106
2	447 Goldfield St.	3	2	1700	CO103
2	594 Leatherwood Dr.	5	2	2750	KO104
1	782 Queen Ln.	4	2	1900	AK102
1	8 Laurel Dr.	4	2	2050	MO100

#### Exercise 3

StayWell has increased the monthly rent of each large property by \$150. Update the monthly rents in the LARGE\_PROPERTY table accordingly.

```
UPDATE LARGE_PROPERTY
SET MONTHLY_RENT = MONTHLY_RENT + 150;
```

#### Output:

OFFICE_NUM	ADDRESS	BDRMS	FLOORS	MONTHLY_RENT	OWNER_NUM
1	105 North Illinois Rd.	3	1	1800	KO104
1	30 West Thomas Rd.	3	1	1550	BU106
2	447 Goldfield St.	3	2	1850	CO103
2	594 Leatherwood Dr.	5	2	2900	KO104
1	782 Queen Ln.	4	2	2050	AK102
1	8 Laurel Dr.	4	2	2200	MO100

#### **Exercise 4**

After increasing the monthly rent of each large property by \$150 (Exercise 3), StayWell decides to decrease the monthly rent of any property whose monthly fee is more than \$1750 by 1 percent. Update the monthly rents in the LARGE\_PROPERTY table accordingly.

```
UPDATE LARGE_PROPERTY
   SET MONTHLY_RENT = MONTHLY_RENT * 0.99
   WHERE MONTHLY_RENT > 1750;
```

OFFICE_NUM	ADDRESS	BDRMS	<b>FLOORS</b>	MONTHLY_RENT	OWNER_NUM
1	105 North Illinois Rd.	3	1	1782	KO104
1	30 West Thomas Rd.	3	1	1550	BU106
2	447 Goldfield St.	3	2	1831.5	CO103
2	594 Leatherwood Dr.	5	2	2871	KO104
1	782 Queen Ln.	4	2	2029.5	AK102
1	8 Laurel Dr.	4	2	2178	MO100



Insert a row into the LARGE\_PROPERTY table for a new property.

```
INSERT INTO LARGE_PROPERTY
VALUES (1, '2643 Lugsi Dr', 3, 2, 775, 'MA111');
```

#### Output:

OFFICE_NUM	ADDRESS	BDRMS	<b>FLOORS</b>	MONTHLY_RENT	OWNER_NUM
1	105 North Illinois Rd.	3	1	1782	KO104
1	2643 Lugsi Dr	3	2	775	MA111
1	30 West Thomas Rd.	3	1	1550	BU106
2	447 Goldfield St.	3	2	1831.5	CO103
2	594 Leatherwood Dr.	5	2	2871	KO104
1	782 Queen Ln.	4	2	2029.5	AK102
1	8 Laurel Dr.	4	2	2178	MO100

#### Exercise 6

Delete all properties in the LARGE\_PROPERTY table for which the owner number is BU106.

```
DELETE FROM LARGE_PROPERTY
  WHERE OWNER_NUM = 'BU106';
```

#### Output:

OFFICE_NUM	ADDRESS	BDRMS	FLOORS	MONTHLY_RENT	OWNER_NUM
1	105 North Illinois Rd.	3	1	1782	KO104
1	2643 Lugsi Dr	3	2	775	MA111
2	447 Goldfield St.	3	2	1831.5	CO103
2	594 Leatherwood Dr.	5	2	2871	KO104
1	782 Queen Ln.	4	2	2029.5	AK102
1	8 Laurel Dr.	4	2	2178	MO100

#### **Exercise 7**

Change the bedrooms value in the LARGE\_PROPERTY table to null for the property managed by Columbia City with the address 105 North Illinois Rd.

```
UPDATE LARGE_PROPERTY
   SET BDRMS = NULL
   WHERE ADDRESS = '105 North Illinois Rd.';
```

OFFICE_NUM	ADDRESS	BDRMS	FLOORS	MONTHLY_RENT	OWNER_NUM
1	105 North Illinois Rd.	NULL	1	1782	KO104
1	2643 Lugsi Dr	3	2	775	MA111
2	447 Goldfield St.	3	2	1831.5	CO103
2	594 Leatherwood Dr.	5	2	2871	KO104
1	782 Queen Ln.	4	2	2029.5	AK102
1	8 Laurel Dr.	4	2	2178	MO100



Add to the LARGE\_PROPERTY table a new character column named OCCUPIED that is one character in length. Set the value for the OCCUPIED column on all rows to Y.

```
ALTER TABLE LARGE_PROPERTY
ADD OCCUPIED CHAR(1);

UPDATE LARGE_PROPERTY
SET OCCUPIED = 'Y';
```

#### Output:

OFFICE_NUM	ADDRESS	<b>BDRMS</b>	<b>FLOORS</b>	MONTHLY_RENT	OWNER_NUM	OCCUPIED
1	1 105 North Illinois Rd.		1	1782	KO104	Υ
1	2643 Lugsi Dr	3	2	775	MA111	Υ
2	447 Goldfield St.	3	2	1831.5	CO103	Υ
2	594 Leatherwood Dr.	5	2	2871	KO104	Υ
1	1 782 Queen Ln.		2	2029.5	AK102	Υ
1	8 Laurel Dr.	4	2	2178	MO100	Υ

#### Exercise 9

Change the OCCUPIED column in the LARGE\_PROPERTY table to N for office 2.

```
UPDATE LARGE_PROPERTY
SET OCCUPIED = 'N'
WHERE OFFICE_NUM = 2;
```

#### **Output:**

OFFICE_NUM	ADDRESS	BDRMS	FLOORS	MONTHLY_RENT	OWNER_NUM	OCCUPIED
1	105 North Illinois Rd.	NULL	1	1782	KO104	Υ
1	2643 Lugsi Dr	3	2	775	MA111	Υ
2	447 Goldfield St.	3	2	1831.5	CO103	N
2	594 Leatherwood Dr.	5	2	2871	KO104	N
1	782 Queen Ln.	4	2	2029.5	AK102	Υ
1	8 Laurel Dr.	4	2	2178	MO100	Υ

#### **Exercise 10**

Change the MONTHLY\_RENT column in the LARGE\_PROPERTY table to reject nulls.

ALTER TABLE LARGE\_PROPERTY

MODIFY MONTHLY\_RENT DECIMAL(6, 2) NOT NULL;

#### **Output:**

Field	Туре	Null	Key	Default	Extra
OFFICE_NUM	decimal(2,0)	YES		NULL	
ADDRESS	char(25)	NO	PRI	NULL	
BDRMS	decimal(2,0)	YES		NULL	
FLOORS	decimal(2,0)	YES		NULL	
MONTHLY_RENT	decimal(6,2)	NO		NULL	
OWNER_NUM	char(5)	YES		NULL	
OCCUPIED	char(1)	YES		NULL	

#### **Exercise 11**

Delete the LARGE\_PROPERTY table from the database.

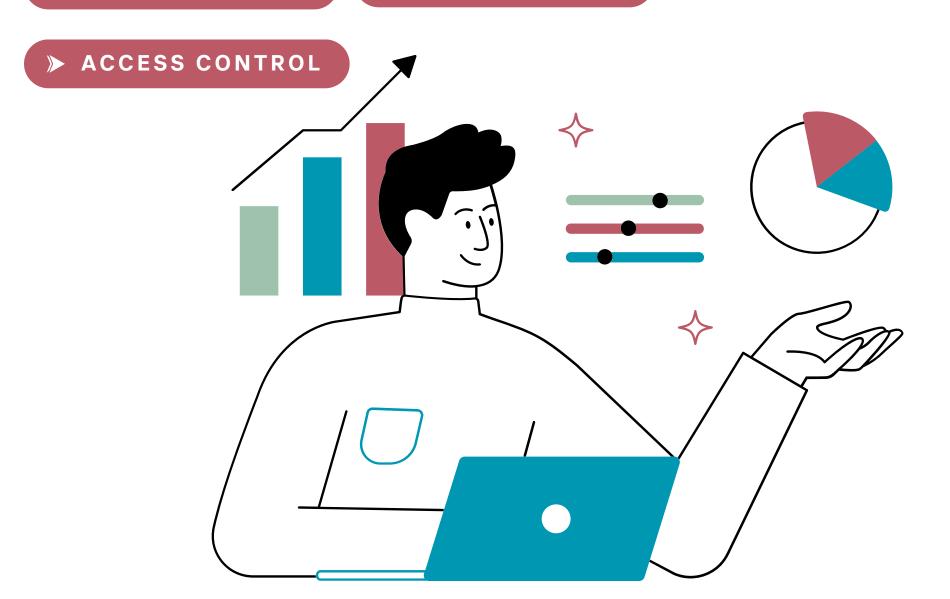
DROP TABLE LARGE\_PROPERTY;



### Database Administration

> VIEWS & INDEXES

**▶** DATA INTEGRITY



#### Objective

This section focuses on essential database administration skills, moving beyond basic DML and DDL to manage the database environment itself. The exercises demonstrate the **creation and use of views** to simplify data access and enhance security. The work also covers the implementation of robust security protocols through **granting and revoking user privileges**, as well as improving database performance by **creating and managing indexes**. Finally, it highlights the enforcement of data integrity through foreign key and check constraints.

#### **Exercise 1a**

Create a view named SMALL\_PROPERTY. It consists of the property ID, office number, bedrooms, floor, monthly rent, and owner number for every property whose square footage is less than 1,250 square feet.

```
CREATE VIEW SMALL_PROPERTY AS
    (SELECT PROPERTY_ID, OFFICE_NUM, BDRMS, FLOORS,
MONTHLY_RENT, OWNER_NUM
     FROM PROPERTY
     WHERE (SQR_FT < 1250)
);</pre>
```

PROPERTY_ID	OFFICE_NUM	BDRMS	FLOORS	MONTHLY_RENT	OWNER_NUM
3	1	2	1	1200	BI109
5	1	2	1	1160	SI105
9	2	2	1	1050	PA101
11	2	2	1	1100	JO110

#### **Exercise 1b**

Write and execute the command to retrieve the office number, property ID, and monthly rent for every property in the SMALL\_PROPERTY view with a monthly rent of \$1150 or more.

```
SELECT OFFICE_NUM, PROPERTY_ID, MONTHLY_RENT
FROM SMALL_PROPERTY
WHERE MONTHLY_RENT >= 1150;
```

#### Output:

OFFICE_NUM	PROPERTY_ID	MONTHLY_RENT
1	3	1200
1	5	1160

#### **Exercise 2a**

Create a view named PROPERTY\_OWNERS. It consists of the property ID, office number, square footage, bedrooms, floors, monthly rent, and owner's last name for every property in which the number of bedrooms is three.

```
CREATE VIEW PROPERTY_OWNERS AS
    (SELECT P.PROPERTY_ID, P.OFFICE_NUM, P.SQR_FT,
P.BDRMS, P.FLOORS, P.MONTHLY_RENT, O.LAST_NAME
        FROM PROPERTY P, OWNER O
        WHERE P.OWNER_NUM = O.OWNER_NUM AND P.BDRMS = 3
);
```

#### Output:

PROPERTY_	ID	OFFICE_NUM	SQR_FT	BDRMS	<b>FLOORS</b>	MONTHLY_RENT	LAST_NAME
	1	1	1600	3	1	1400	Burke
	4	1	1750	3	1	1650	Kowalczyk
	7	2	1675	3	2	1700	Cole
	10	2	1300	3	1	1600	Lopez
	12	2	1400	3	2	1700	Redman

#### Exercise 2b

Write and execute the command to retrieve the property ID, office number, monthly rent, square footage, and owner's last name for every property in the PROPERTY\_OWNERS view with a monthly rent of less than \$1675.

```
SELECT PROPERTY_ID, OFFICE_NUM, MONTHLY_RENT, SQR_FT,
LAST_NAME
FROM PROPERTY_OWNERS
WHERE MONTHLY_RENT < 1675;
```

PROPERTY_ID	OFFICE_NUM	MONTHLY_RENT	SQR_FT	LAST_NAME
1	1	1400	1600	Burke
4	1	1650	1750	Kowalczyk
10	2	1600	1300	Lopez





Create a view named MONTHLY\_RENTS. It consists of two columns: The first is the number of bedrooms, and the second is the average monthly rent for all properties in the PROPERTY table that have that number of bedrooms. Use AVERAGE\_RENT as the column name for the average monthly rent. Group and order the rows by number of bedrooms.

```
CREATE VIEW MONTHLY_RENTS AS

(SELECT BDRMS, AVG(MONTHLY_RENT) AS AVERAGE_RENT

FROM PROPERTY

GROUP BY BDRMS

ORDER BY BDRMS

);
```

#### Output:

BDRMS	AVERAGE_RENT	
2	1127.5	
3	1610	
4	1975	
5	2750	

#### **Exercise 4a**

Write commands to grant the following privileges. User Oliver must be able to retrieve data from the PROPERTY table.

GRANT SELECT ON PROPERTY TO Oliver;

Users Crandall and Perez must be able to add new owners and properties to the database.

```
GRANT INSERT ON OWNER TO Crandall, Perez;
GRANT INSERT ON PROPERTY TO Crandall, Perez;
```

#### **Exercise 4c**

Users Johnson and Klein must be able to change the monthly rent of any unit.

```
GRANT UPDATE (MONTHLY_RENT) ON PROPERTY TO Johnson,
Klein;
```

#### **Exercise 4d**

All users must be able to retrieve the office number, monthly rent, and owner number for every property.

```
GRANT SELECT (OFFICE_NUM, MONTHLY_RENT, OWNER_NUM) ON PROPERTY TO PUBLIC;
```

#### **Exercise 4e**

User Klein must be able to add and delete service categories.

```
GRANT INSERT, DELETE ON SERVICE_CATEGORY TO Klein;
```

#### **Exercise 4f**

User Adams must be able to create an index on the SERVICE\_REQUEST table.

```
GRANT INDEX ON SERVICE_REQUEST TO Adams;
```

#### **Exercise 4g**

Users Adams and Klein must be able to change the structure of the PROPERTY table.

GRANT ALTER ON PROPERTY TO Adams, Klein;

#### Exercise 4h

User Klein must have all privileges on the OFFICE, OWNER, and PROPERTY tables.

GRANT ALL PRIVILEGES ON OFFICE, OWNER, PROPERTY TO Klein;

#### **Exercise 5**

Write the command to revoke all privileges from user Adams.

REVOKE ALL PRIVILEGES ON PROPERTY FROM Adams; REVOKE INDEX ON SERVICE\_REQUEST FROM Adams;

#### Exercise 6a

Create an index named OWNER\_INDEX1 on the STATE column in the OWNER table.

CREATE INDEX OWNER\_INDEX1 ON OWNER (STATE);

#### Exercise 6b

Create an index named OWNER\_INDEX2 on the LAST\_NAME column in the OWNER table.

CREATE INDEX OWNER\_INDEX2 ON OWNER (LAST\_NAME);

#### Exercise 6c

Create an index named OWNER\_INDEX3 on the STATE and CITY columns in the OWNER table. List the states in descending order.

CREATE INDEX OWNER\_INDEX3 ON OWNER (STATE DESC, CITY);

#### Exercise 7

Delete the OWNER\_INDEX 3 index from the OWNER table.

DROP INDEX OWNER\_INDEX3 ON OWNER;

#### Exercise 8a

List every column in the PROPERTY table and its associated data type.

```
SELECT COLUMN_NAME, DATA_TYPE

FROM INFORMATION_SCHEMA.COLUMNS

WHERE TABLE_SCHEMA = 'StayWell' AND TABLE_NAME =

'PROPERTY';
```

COLUMN_NAME	DATA_TYPE
ADDRESS	varchar
BDRMS	tinyint
FLOORS	tinyint
MONTHLY_RENT	smallint
OFFICE_NUM	tinyint
OWNER_NUM	char
PROPERTY_ID	smallint
SQR_FT	smallint

#### **Exercise 8b**

List every table that contains a column named OWNER\_NUM.

```
SELECT TABLE_NAME
    FROM INFORMATION_SCHEMA.COLUMNS
    WHERE COLUMN_NAME = 'OWNER_NUM' AND TABLE_SCHEMA
= 'StayWell';
```

#### Output:

TABLE_NAME
OWNER
PROPERTY
small_property



#### Exercise 9

Add the OWNER\_NUM column as a foreign key in the PROPERTY table.

```
ALTER TABLE PROPERTY

ADD FOREIGN KEY (OWNER_NUM) REFERENCES OWNER

(OWNER_NUM);
```

#### Output:

PROPERTY_ID	OFFICE_NUM	ADDRESS	SQR_FT	<b>BDRMS</b>	<b>FLOORS</b>	MONTHLY_RENT	OWNER_NUM
1	1	30 West Thomas Rd.	1600	3	1	1400	BU106
2	1	782 Queen Ln.	2100	4	2	1900	AK102
3	1	9800 Sunbeam Ave.	1005	2	1	1200	BI109
4	1	105 North Illinois Rd.	1750	3	1	1650	KO104
5	1	887 Vine Rd.	1125	2	1	1160	SI105
6	1	8 Laurel Dr.	2125	4	2	2050	MO100
7	2	447 Goldfield St.	1675	3	2	1700	CO103
8	2	594 Leatherwood Dr.	2700	5	2	2750	KO104
9	2	504 Windsor Ave.	700	2	1	1050	PA101
10	2	891 Alton Dr.	1300	3	1	1600	LO108
11	2	9531 Sherwood Rd.	1075	2	1	1100	JO110
12	2	2 Bow Ridge Ave.	1400	3	2	1700	RE107

#### **Exercise 10**

Ensure that the only legal values for the BDRMS column in the PROPERTY table are 1, 2, or 3.

```
ALTER TABLE PROPERTY

ADD CHECK (BDRMS IN (1, 2, 3));
```

Functions, Procedures, & Triggers

**▶** PROCEDURAL PROGRAMMING

**▶** ADVANCED SQL

**▶** CURSORS



#### **Objective**

This final section demonstrates an understanding of procedural programming concepts within a SQL environment. It showcases the ability to **create and execute stored procedures** for a wide range of tasks, including single-row data retrieval, bulk data manipulation, and multi-row processing using cursors. The work also highlights the use of built-in functions to format and calculate data directly within queries, proving a sophisticated level of control over database operations.

#### Exercise 1

List the owner number, first name, and last name for all owners. The first name should appear in uppercase letters and the last name should appear in lowercase letters.

SELECT OWNER\_NUM, UPPER(FIRST\_NAME) AS
UPPER\_FIRST\_NAME, LOWER(LAST\_NAME) AS LOWER\_LAST\_NAME
FROM OWNER;

OWNER_NUM	UPPER_FIRST_NAME	LOWER_LAST_NAME	
AK102	CEYDA	aksoy	
BI109	NICOLE	bianchi	
BU106	ERNEST	burke	
CO103	MEERAB	cole	
JO110	AMMARAH	jones	
KO104	JAKUB	kowalczyk	
LO108	JANINE	lopez	
MO100	ELLE-MAY	moore	
PA101	MAKESH	patel	
RE107	SETH	redman	
SI105	HAYDON	sims	



List the owner number and last name for all owners located in the city of Seattle. This query should ignore case.

```
SELECT OWNER_NUM, LAST_NAME

FROM OWNER

WHERE UPPER(CITY) = 'SEATTLE';
```

#### Output:

OWNER_NUM	LAST_NAME
AK102	Aksoy
JO110	Jones
PA101	Patel
RE107	Redman

#### Exercise 3

StayWell is offering a monthly discount for residents who pay their rent on a quarterly basis. The discount is 1.75 percent of the monthly fee. For each property, list the office number, address, owner number, owner's last name, monthly rent, and discount. The discount should be rounded to the nearest dollar.

```
P.OFFICE_NUM,
P.ADDRESS,
O.LAST_NAME,
P.MONTHLY_RENT,
ROUND(P.MONTHLY_RENT * 0.0175, 0) AS MONTHLY_DISCOUNT
FROM PROPERTY P
JOIN OWNER O ON P.OWNER_NUM = 0.0WNER_NUM;
```

#### Output:

OFFICE_NUM	ADDRESS	LAST_NAME	MONTHLY_RENT	MONTHLY_DISCOUNT
1	30 West Thomas Rd.	Burke	1400	25
1	782 Queen Ln.	Aksoy	1900	33
1	9800 Sunbeam Ave.	Bianchi	1200	21
1	105 North Illinois Rd.	Kowalczyk	1650	29
1	887 Vine Rd.	Sims	1160	20
1	8 Laurel Dr.	Moore	2050	36
2	447 Goldfield St.	Cole	1700	30
2	594 Leatherwood Dr.	Kowalczyk	2750	48
2	504 Windsor Ave.	Patel	1050	18
2	891 Alton Dr.	Lopez	1600	28
2	9531 Sherwood Rd.	Jones	1100	19
2	2 Bow Ridge Ave.	Redman	1700	30

#### **Exercise 4a**

Write a procedure to obtain the first and last name of an owner.

```
DELIMITER //
CREATE PROCEDURE GET_OWNER_NAME (IN I_OWNER_NUM CHAR(5))
BEGIN

DECLARE V_FIRST_NAME VARCHAR(50);
DECLARE V_LAST_NAME VARCHAR(50);
SELECT FIRST_NAME, LAST_NAME
INTO V_FIRST_NAME, V_LAST_NAME
FROM OWNER
WHERE OWNER_NUM = I_OWNER_NUM;
SELECT I_OWNER_NUM, V_FIRST_NAME, V_LAST_NAME;
END //
DELIMITER;
```

#### **Exercise 4b**

Write a procedure to obtain property and owner details for a specific property.

```
DELIMITER //
CREATE PROCEDURE GET_PROPERTY_AND_OWNER_DETAILS (IN I_PROPERTY_ID INT)
BEGIN
  DECLARE V OFFICE NUM INT;
  DECLARE V_OFFICE_ADDRESS VARCHAR(100);
  DECLARE V PROPERTY ADDRESS VARCHAR(100);
  DECLARE V_OWNER_NUM CHAR(5);
  DECLARE V FIRST NAME VARCHAR(50);
  DECLARE V_LAST_NAME VARCHAR(50);
  SELECT
     P.OFFICE_NUM,
     OFC.ADDRESS,
     P.ADDRESS,
     P.OWNER_NUM,
     OW.FIRST NAME,
     OW.LAST NAME
  INTO
     V_OFFICE_NUM,
     V_OFFICE_ADDRESS,
     V_PROPERTY_ADDRESS,
     V_OWNER_NUM,
     V_FIRST_NAME,
     V_LAST_NAME
  FROM
      PROPERTY P
   JOIN
      OFFICE OFC ON P.OFFICE_NUM = OFC.OFFICE_NUM
   JOIN
      OWNER OW ON P.OWNER_NUM = OW.OWNER_NUM
  WHERE
      P.PROPERTY_ID = I_PROPERTY_ID;
  SELECT I_PROPERTY_ID, V_OFFICE_NUM, V_OFFICE_ADDRESS,
V_PROPERTY_ADDRESS, V_OWNER_NUM, V_FIRST_NAME, V_LAST_NAME;
END //
DELIMITER;
```

#### **Exercise 4c**

Write a procedure to add a row to the OWNER table.

```
DELIMITER //
CREATE PROCEDURE ADD_OWNER_ROW ()
BEGIN
    INSERT INTO OWNER (OWNER_NUM, LAST_NAME, FIRST_NAME,
ADDRESS, CITY, STATE, ZIP_CODE)
    VALUES ('AD123', 'Dempsey', 'Allison', '123 Address
Dr', 'St Petersburg', 'FL', '33710');
END //
DELIMITER;
```

#### **Exercise 4d**

Write a procedure to change the last name of an owner.

```
DDELIMITER //
CREATE PROCEDURE CHG_OWNER_LAST_NAME (
        IN I_OWNER_NUM CHAR(5),
        IN I_LAST_NAME VARCHAR(50)
)
BEGIN
        UPDATE OWNER
        SET LAST_NAME = I_LAST_NAME
        WHERE OWNER_NUM = I_OWNER_NUM;
END //
DELIMITER;
```



#### **Exercise 4e**

Write a procedure to delete an owner.

```
DELIMITER //
CREATE PROCEDURE DEL_OWNER (IN I_OWNER_NUM CHAR(5))
BEGIN
    DELETE FROM OWNER
    WHERE OWNER_NUM = I_OWNER_NUM;
END //
DELIMITER;
```

#### **Exercise 5**

Write a procedure to retrieve multiple properties whose square footage is equal to the square footage stored in I SQR FT.

DELIMITER // CREATE PROCEDURE GET\_PROPERTIES\_BY\_SQR\_FT (IN I\_SQR\_FT INT) BEGIN DECLARE DONE INT DEFAULT FALSE; DECLARE V\_OFFICE\_NUM INT; DECLARE V\_OFFICE\_ADDRESS VARCHAR(100); DECLARE V\_PROPERTY\_ADDRESS VARCHAR(100); DECLARE V\_MONTHLY\_RENT DECIMAL(10, 2); DECLARE V\_OWNER\_NUM CHAR(5); DECLARE PROPGROUP CURSOR FOR SELECT P.OFFICE\_NUM, OFC.ADDRESS, P.ADDRESS, P.MONTHLY\_RENT, P.OWNER\_NUM FROM PROPERTY P JOIN OFFICE OFC ON P.OFFICE\_NUM = OFC.OFFICE\_NUM WHERE  $P.SQR_FT = I_SQR_FT;$ DECLARE CONTINUE HANDLER FOR NOT FOUND SET DONE = TRUE; OPEN PROPGROUP; read\_loop: LOOP FETCH PROPGROUP INTO V\_OFFICE\_NUM, V\_OFFICE\_ADDRESS, V\_PROPERTY\_ADDRESS, V\_MONTHLY\_RENT, V\_OWNER\_NUM; IF DONE THEN LEAVE read\_loop; END IF; SELECT V\_OFFICE\_NUM, V\_OFFICE\_ADDRESS, V\_PROPERTY\_ADDRESS, V\_MONTHLY\_RENT, V\_OWNER\_NUM; END LOOP; CLOSE PROPGROUP; END // DELIMITER;

28



## Key Business Takeaways

Enhanced Data-Driven Decision Making

The database allows for complex queries and analysis that would be difficult with manual records. For example, management can easily identify properties with specific attributes, track service requests, and analyze rental trends, leading to more informed business decisions.

Improved Operational Efficiency

By centralizing property, owner, and resident data in a relational database, the company can move away from manual record-keeping and direct communication. This streamlined approach makes information easily accessible and provides a foundation for future digital platforms like online booking systems and mobile apps.



Enforced Data Integrity and Quality

The implementation of integrity constraints, such as primary and foreign keys, ensures the accuracy and consistency of data. This prevents invalid data entry and maintains reliable records for all business operations.

Scalability and Future-Proofing

The relational database model is designed to grow with the business. It provides a flexible and organized structure that can handle an increase in properties, owners, and service requests, supporting StayWell's long-term expansion goals.

