USIU AFRICA United States International University-Africa

DATABASE SYSTEMS

Presented by Prof. Elisha T. O. Omulo

USIU AFRICA

WEEK 8 AGENDA

- **Single - and multiple-table queries using SQL commands.**
- **Three types of join commands and their SQL implementations.**
- **Writing Subqueries**
- **Referential integrity using SQL.**
- **Database triggers and stored procedures.**

Course Textbook: Carlos Coronel, Steven Morris, Peter Rob and Keeley Crockett Database Principles: Fundamentals of Design, Implementation, and Management, 14th Edition, 2022, ISBN-13978-0357673034.

USIU AFRICA

Single - and multiple-table queries using SQL commands.

We will highlight a number of SQL commands for both single and multiple tables.

SQL command

```
SELECT columnlist
FROM tablelist
[WHERE conditionlist ]
[GROUP BY columnlist ]
[HAVING conditionlist ]
[ORDER BY columnlist [ASC | DESC] ] ;
```

SELECT is a powerful tool that transforms data into information.
Select partial table contents, restricting the rows to be included in the output.
The WHERE clause adds conditional restrictions to the SELECT statement.

USIU AFRICA

Task: Discuss the SELECT command format.

Single - and multiple-table queries using SQL commands.

SQL commands for both single and multiple tables

- **SQL statements are case insensitive, so letters can be typed in either upper or lower case, but as exception to this rule, the literal character data must be typed exactly as it appears eg. a person's surname 'Mwangi' is searched using string 'Mwangi'.**
- **Although SQL is free-format, an SQL statement or set of statements is more readable if indentation and lineation are used.**

So :

- **Each clause in a statement should begin on a new line;**
- **The beginning of each clause should line up with the beginning of other clauses;**
- **If a clause has several parts, they should each appear on a separate line and be indented under the start of the clause to show the relationship.**

Task: Discuss the formatting of the SELECT statements.

Single - and multiple-table queries using SQL commands.

SQL commands for both single and multiple tables

```
SELECT P_DESCRIPT, P_INDATE, P_PRICE, V_CODE
FROM PRODUCT
WHERE V_CODE = 21344;
```

- **The SELECT statement retrieves all rows that match the specified condition(s)—also known as the conditional criteria, specified in the WHERE clause.**
- **The conditionlist in the WHERE clause of the SELECT statement is represented by one or more conditional expressions, separated by logical operators.**

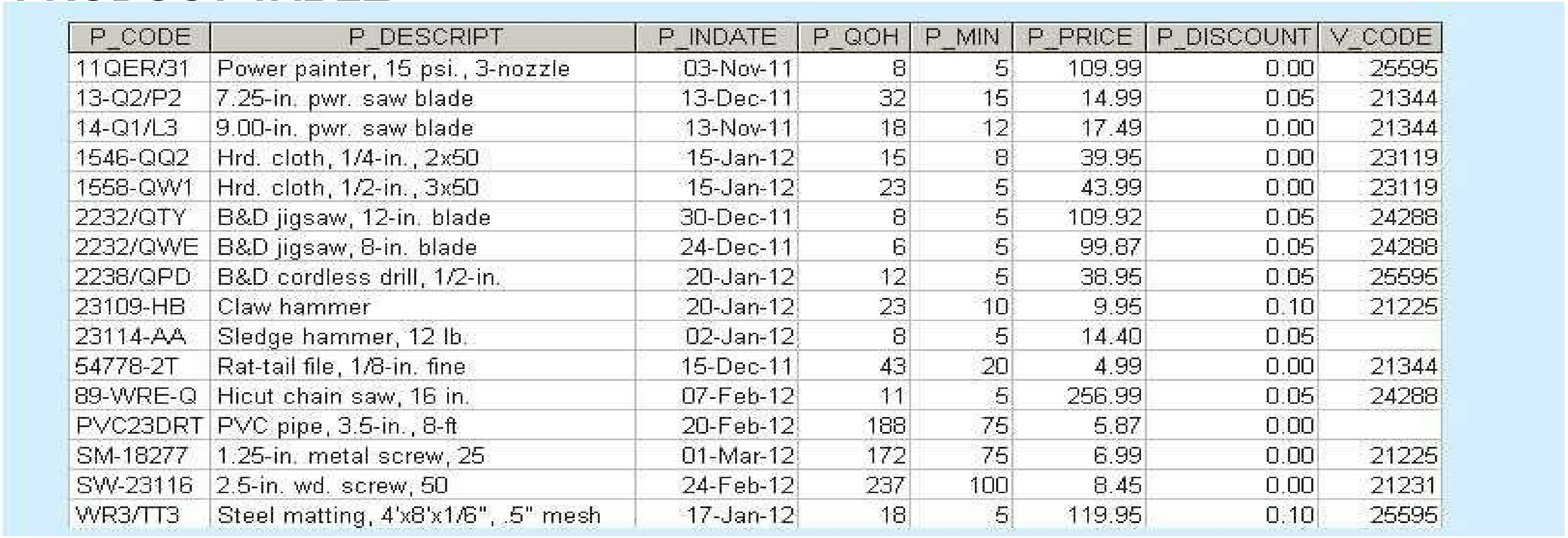
Task: Discuss the formatting of the SELECT statements.

USIU AFRICA

Single - and multiple-table queries using SQL commands.

SQL commands for both single and multiple tables

PRODUCT TABLE



P_CODE	P_DESCRIPT	P_INDATE	P_QOH	P_MIN	P_PRICE	P_DISCOUNT	V_CODE
11QER/31	Power painter, 15 psi., 3-nozzle	03-Nov-11	8	5	109.99	0.00	25595
13-Q2/P2	7.25-in. pwr. saw blade	13-Dec-11	32	15	14.99	0.05	21344
14-Q1/L3	9.00-in. pwr. saw blade	13-Nov-11	18	12	17.49	0.00	21344
1546-QQ2	Hrd. cloth, 1/4-in., 2x50	15-Jan-12	15	8	39.95	0.00	23119
1558-QW1	Hrd. cloth, 1/2-in., 3x50	15-Jan-12	23	5	43.99	0.00	23119
2232/QTY	B&D jigsaw, 12-in. blade	30-Dec-11	8	5	109.92	0.05	24288
2232/QWE	B&D jigsaw, 8-in. blade	24-Dec-11	6	5	99.87	0.05	24288
2238/QPD	B&D cordless drill, 1/2-in.	20-Jan-12	12	5	38.95	0.05	25595
23109-HB	Claw hammer	20-Jan-12	23	10	9.95	0.10	21225
23114-AA	Sledge hammer, 12 lb.	02-Jan-12	8	5	14.40	0.05	
54778-2T	Rat-tail file, 1/8-in. fine	15-Dec-11	43	20	4.99	0.00	21344
89-WRE-Q	Hicut chain saw, 16 in.	07-Feb-12	11	5	256.99	0.05	24288
PVC23DRT	PVC pipe, 3.5-in., 8-ft	20-Feb-12	188	75	5.87	0.00	
SM-18277	1.25-in. metal screw, 25	01-Mar-12	172	75	6.99	0.00	21225
SW-23116	2.5-in. wd. screw, 50	24-Feb-12	237	100	8.45	0.00	21231
WR3/TT3	Steel matting, 4'x8'x1/6", .5" mesh	17-Jan-12	18	5	119.95	0.10	25595

Task: Formulate SELECT commands for the PRODUCT table.

Single - and multiple-table queries using SQL commands.

SQL commands for both single and multiple tables

VENDOR-PRODUCT TABLES

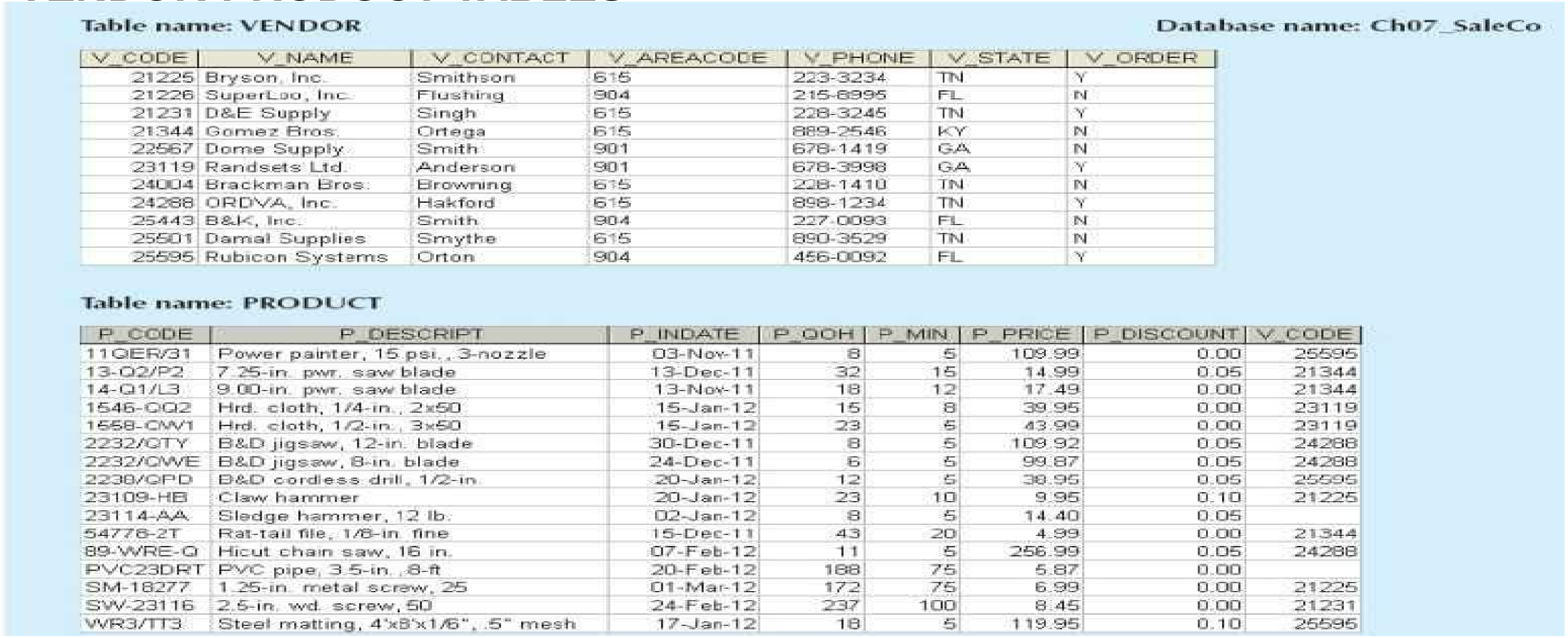


Table name: VENDOR **Database name: Ch07_SaleCo**

V_CODE	V_NAME	V_CONTACT	V_AREACODE	V_PHONE	V_STATE	V_ORDER
21225	Bryson, Inc.	Smithson	615	223-3234	TN	Y
21226	SuperLoo, Inc.	Flushing	904	215-8995	FL	N
21231	D&E Supply	Singh	615	228-3245	TN	Y
21344	Gomez Bros.	Ortega	615	889-2546	KY	N
22567	Dome Supply	Smith	901	678-1419	GA	N
23119	Randsets Ltd.	Anderson	901	678-3998	GA	Y
24004	Brackman Bros.	Browning	615	228-1410	TN	N
24288	ORDVA, Inc.	Hakford	615	898-1234	TN	Y
25443	B&K, Inc.	Smith	904	227-0093	FL	N
25501	Damal Supplies	Smythe	615	890-3529	TN	N
25595	Rubicon Systems	Orton	904	456-0092	FL	Y

Table name: PRODUCT

P_CODE	P_DESCRIPT	P_INDATE	P_QOH	P_MIN	P_PRICE	P_DISCOUNT	V_CODE
11QER/31	Power painter, 15 psi., 3-nozzle	03-Nov-11	8	5	109.99	0.00	25595
13-Q2/P2	7.25-in. pwr. saw blade	13-Dec-11	32	15	14.99	0.05	21344
14-Q1/L3	9.00-in. pwr. saw blade	13-Nov-11	18	12	17.49	0.00	21344
1546-QQ2	Hrd. cloth, 1/4-in., 2x50	15-Jan-12	15	8	39.95	0.00	23119
1558-QW1	Hrd. cloth, 1/2-in., 3x50	15-Jan-12	23	5	43.99	0.00	23119
2232/QTY	B&D jigsaw, 12-in. blade	30-Dec-11	8	5	109.92	0.05	24288
2232/QWE	B&D jigsaw, 8-in. blade	24-Dec-11	6	5	99.87	0.05	24288
2238/QPD	B&D cordless drill, 1/2-in.	20-Jan-12	12	5	38.95	0.05	25595
23109-HB	Claw hammer	20-Jan-12	23	10	9.95	0.10	21225
23114-AA	Sledge hammer, 12 lb.	02-Jan-12	8	5	14.40	0.05	
54778-2T	Rat-tail file, 1/8-in. fine	15-Dec-11	43	20	4.99	0.00	21344
89-WRE-Q	Hicut chain saw, 16 in.	07-Feb-12	11	5	256.99	0.05	24288
PVC23DRT	PVC pipe, 3.5-in., 8-ft	20-Feb-12	188	75	5.87	0.00	
SM-18277	1.25-in. metal screw, 25	01-Mar-12	172	75	6.99	0.00	21225
SW-23116	2.5-in. wd. screw, 50	24-Feb-12	237	100	8.45	0.00	21231
WR3/TT3	Steel matting, 4'x8'x1/6", .5" mesh	17-Jan-12	18	5	119.95	0.10	25595

Task: Formulate SELECT commands for both tables.

Single - and multiple-table queries using SQL commands.

SQL commands for both single and multiple tables

```
SELECT P_DESCRIPT, P_INDATE, P_PRICE, V_CODE
FROM PRODUCT
WHERE V_CODE <> 21344;

SELECT P_DESCRIPT, P_QOH, P_MIN, P_PRICE
FROM PRODUCT
WHERE P_PRICE <= 10;

SELECT P_CODE, P_DESCRIPT, P_QOH, P_MIN, P_PRICE
FROM PRODUCT
WHERE P_CODE < '1558-QW1';

SELECT P_DESCRIPT, P_QOH, P_PRICE, P_QOH * P_PRICE
FROM PRODUCT
```

Computed column(*) Exprl1 will be automatically created

Task: Give all the commands above.

USIU AFRICA

Single - and multiple-table queries using SQL commands.

SQL commands for both single and multiple tables

```
SELECT P_DESCRIPT, P_QOH, P_PRICE, P_QOH * P_PRICE
FROM PRODUCT
```

Computed column(*) Exprl1 will be automatically created

Alternatively write as:

```
SELECT P_DESCRIPT, P_QOH, P_PRICE, P_QOH * P_PRICE AS TOTVALUE
FROM PRODUCT;
```

Access version=DATE:

```
SELECT P_CODE, P_INDATE, DATE() - 90 AS CUTDATE
FROM PRODUCT
WHERE P_INDATE <= DATE() - 90;
```

The Oracle version of the same query:

```
SELECT P_CODE, P_INDATE, SYSDATE CUTDATE - 90 AS
FROM PRODUCT
WHERE P_INDATE <= SYSDATE - 90;
```

Task: Give all the commands above.

Single - and multiple-table queries using SQL commands.

SQL commands for both single and multiple tables

```
SELECT P_DESCRIPT, P_INDATE, P_PRICE, V_CODE
FROM PRODUCT
WHERE V_CODE = 21344 OR V_CODE = 24288;

SELECT P_DESCRIPT, P_INDATE, P_PRICE, V_CODE
FROM PRODUCT
WHERE P_PRICE < 50   AND P_INDATE > '15-Jan-2012';

SELECT *
FROM PRODUCT
WHERE NOT (V_CODE = 21344);       NB could have used <>, != depending on implementation

SELECT *
FROM PRODUCT
WHERE P_PRICE BETWEEN 50.00 AND 100.00;
```

Task: Give all the commands above.

USIU AFRICA

Single - and multiple-table queries using SQL commands.

SQL commands for both single and multiple tables

```
SELECT *
FROM PRODUCT
WHERE P_PRICE > 50.00 AND P_PRICE < 100.00; nb if BETWEEN is not supported.

SELECT P_CODE, P_DESCRIPT, V_CODE
FROM PRODUCT
WHERE V_CODE IS NULL;

SELECT P_CODE, P_DESCRIPT, P_INDATE
FROM PRODUCT
WHERE P_INDATE IS NULL;   nb: check a null date entry

SELECT V_NAME, V_CONTACT, V_AREACODE, V_PHONE
FROM VENDOR
WHERE V_CONTACT LIKE 'Neema';  nb find vendor contracts with name 'Neema'
```

Task: Give all the commands above.

USIU AFRICA

Single - and multiple-table queries using SQL commands.

SQL commands for both single and multiple tables

```
SELECT *
FROM PRODUCT
WHERE V_CODE IN (21344, 24288);   nb: V_CODE = 21288 OR 24288

SELECT V_CODE, V_NAME
FROM VENDOR
WHERE V_CODE IN (SELECT V_CODE FROM PRODUCT);    nb inner query

SELECT *
FROM VENDOR
WHERE EXISTS (SELECT * FROM PRODUCT WHERE P_QOH <= P_MIN);  nb subquery when exit products to order

SELECT *
FROM VENDOR
WHERE EXISTS (SELECT * FROM PRODUCT WHERE P_QOH < P_MIN * 2); nb subquery
```

Task: Demonstrate all the commands above.

USIU AFRICA

Single - and multiple-table queries using SQL commands.

SQL commands for both single and multiple tables

```
SELECT P_CODE, P_DESCRIPT, P_INDATE, P_PRICE
FROM PRODUCT
ORDER BY P_PRICE;

SELECT P_CODE, P_DESCRIPT, P_INDATE, P_PRICE
FROM PRODUCT
ORDER BY P_PRICE DESC;

SELECT P_DESCRIPT, V_CODE, P_INDATE, P_PRICE
FROM PRODUCT
WHERE P_INDATE < '21-Jan-2012' AND P_PRICE <= 50.00
ORDER BY V_CODE, P_PRICE DESC;

SELECT DISTINCT V_CODE
FROM PRODUCT;                          nb. To count the vendors for the product
```

Task: Demonstrate all the commands above.

USIU AFRICA

Single - and multiple-table queries using SQL commands.

SQL commands for both single and multiple tables

```
SELECT P_DESCRIPT, P_PRICE, V_NAME, V_CONTACT, V_AREACODE, V_PHONE
FROM PRODUCT, VENDOR
WHERE PRODUCT.V_CODE = VENDOR.V_CODE;    tables are joined

SELECT PRODUCT.P_DESCRIPT, PRODUCT.P_PRICE, VENDOR.V_NAME, VENDOR.V_CONTACT, VENDOR.V_AREACODE, VENDOR.V_PHONE
FROM PRODUCT, VENDOR
WHERE  PRODUCT.V_CODE = VENDOR.V_CODE
ORDER BY PRODUCT.P_PRICE;  nb tables are joined

SELECT P_DESCRIPT, P_PRICE, V_NAME, V_CONTACT, V_AREACODE, V_PHONE
FROM PRODUCT, VENDOR
WHERE PRODUCT.V_CODE = VENDOR.V_CODE
AND P_INDATE > '15-Jan-2012';
```

Task: Demonstrate all the commands above.

USIU AFRICA

Single - and multiple-table queries using SQL commands.

SQL commands for both single and multiple tables

```
SELECT CUS_LNAME, INVOICE.INV_NUMBER, INV_DATE, P_DESCRIPT
FROM CUSTOMER, INVOICE, LINE, PRODUCT
WHERE CUSTOMER.CUS_CODE = INVOICE.CUS_CODE
        AND INVOICE.INV_NUMBER = LINE.INV_NUMBER
        AND LINE.P_CODE = PRODUCT.P_CODE
        AND CUSTOMER.CUS_CODE = 10014
ORDER BY INV_NUMBER;

SELECT P_DESCRIPT, P_PRICE, V_NAME, V_CONTACT, V_AREACODE, V_PHONE
FROM PRODUCT P, VENDOR V
WHERE P.V_CODE = V.V_CODE
ORDER BY P_PRICE;                nb P and V are aliases

SELECT E.EMP_MGR, M.EMP_LNAME, E.EMP_NUM, E.EMP_LNAME
FROM EMP E, EMP M
WHERE E.EMP_MGR=M.EMP_NUM
ORDER BY E.EMP_MGR;
```

Task: Demonstrate all the commands above.

Three types of join commands and their SQL implementations.

- The relational join operation merges rows from two tables and returns the rows
- The join condition will be an equality comparison of the primary key in one table and the related foreign key in the second table.

The inner join

Is the traditional join in which only rows that meet a given criterion are selected.

Join criterion: equality condition (also called a natural join or an equijoin)

or an inequality condition (also called a theta join).

```
SELECT column-list FROM table1 NATURAL JOIN table2
```

```
SELECT CUS_CODE, CUS_LNAME, INV_NUMBER, INV_DATE
FROM CUSTOMER NATURAL JOIN INVOICE;
```

Task: As at least 5 different examples of supertypes and subtypes.

USIU AFRICA

Three types of join commands and their SQL implementations.

- The relational join operation merges rows from two tables and returns the rows
- The join condition will be an equality comparison of the primary key in one table and the related foreign key in the second table.

The outer join

- Returns not only the matching rows but the rows with unmatched attribute values for one table or both tables to be joined.
- It still returns all of the matched records that the inner join returns, plus it returns the unmatched records from one of the tables.
- Outer join produced for tables CUSTOMER and AGENT, gives two cases:

Left outer join yields all of the rows in the CUSTOMER table, including those that do not have a matching value in the AGENT table.

Right outer join yields all of the rows in the AGENT table, including those that do not have matching values in the CUSTOMER table.

Task: Discuss the outer join operator

Three types of join commands and their SQL implementations.

- The relational join operation merges rows from two tables and returns the rows
- The join condition will be an equality comparison of the primary key in one table and the related foreign key in the second table.

The cross join

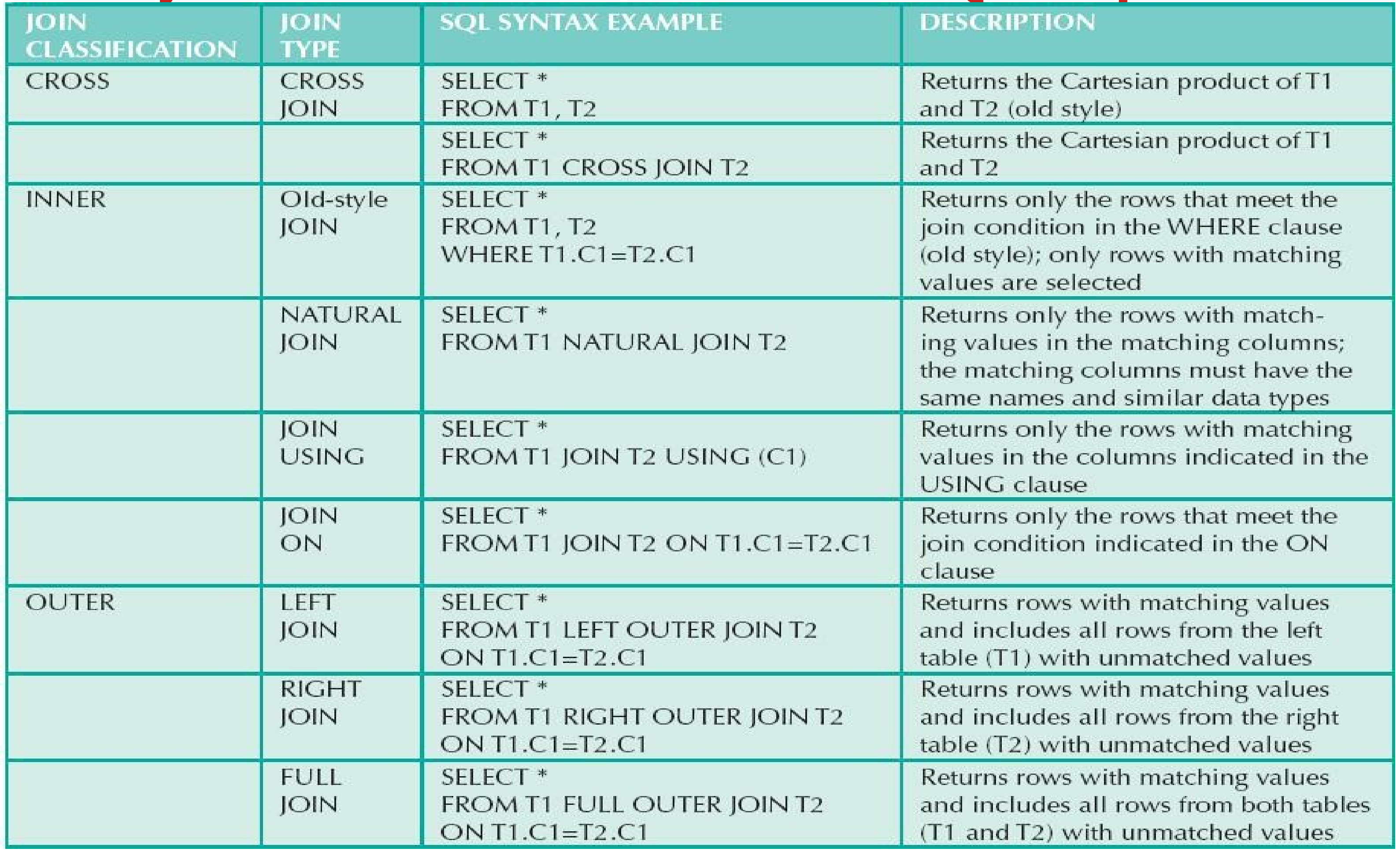
A relational product (also known as the Cartesian product) of two tables.

```
SELECT column-list FROM table1 CROSS JOIN table2
```

```
SELECT * FROM INVOICE CROSS JOIN LINE;
```

Task: Discuss the outer join operator

Three types of join commands and their SQL implementations.



JOIN CLASSIFICATION	JOIN TYPE	SQL SYNTAX EXAMPLE	DESCRIPTION
CROSS	CROSS JOIN	SELECT * FROM T1, T2	Returns the Cartesian product of T1 and T2 (old style)
		SELECT * FROM T1 CROSS JOIN T2	Returns the Cartesian product of T1 and T2
INNER	Old-style JOIN	SELECT * FROM T1, T2 WHERE T1.C1=T2.C1	Returns only the rows that meet the join condition in the WHERE clause (old style); only rows with matching values are selected
	NATURAL JOIN	SELECT * FROM T1 NATURAL JOIN T2	Returns only the rows with matching values in the matching columns; the matching columns must have the same names and similar data types
	JOIN USING	SELECT * FROM T1 JOIN T2 USING (C1)	Returns only the rows with matching values in the columns indicated in the USING clause
	JOIN ON	SELECT * FROM T1 JOIN T2 ON T1.C1=T2.C1	Returns only the rows that meet the join condition indicated in the ON clause
OUTER	LEFT JOIN	SELECT * FROM T1 LEFT OUTER JOIN T2 ON T1.C1=T2.C1	Returns rows with matching values and includes all rows from the left table (T1) with unmatched values
	RIGHT JOIN	SELECT * FROM T1 RIGHT OUTER JOIN T2 ON T1.C1=T2.C1	Returns rows with matching values and includes all rows from the right table (T2) with unmatched values
	FULL JOIN	SELECT * FROM T1 FULL OUTER JOIN T2 ON T1.C1=T2.C1	Returns rows with matching values and includes all rows from both tables (T1 and T2) with unmatched values

Task: Discuss how to write queries with the Join command

USIU AFRICA

Writing Subqueries

A subquery is a query (SELECT statement) inside a query.

- **A subquery is normally expressed inside parentheses.**
- **The first query in the SQL statement is known as the outer query.**
- **The query inside the SQL statement is known as the inner query.**
- **The inner query is executed first.**
- **The output of an inner query is used as the input for the outer query.**
- **The entire SQL statement is sometimes referred to as a nested query.**

Task: Write subqueries.

USIU
AFRICA

Writing Subqueries

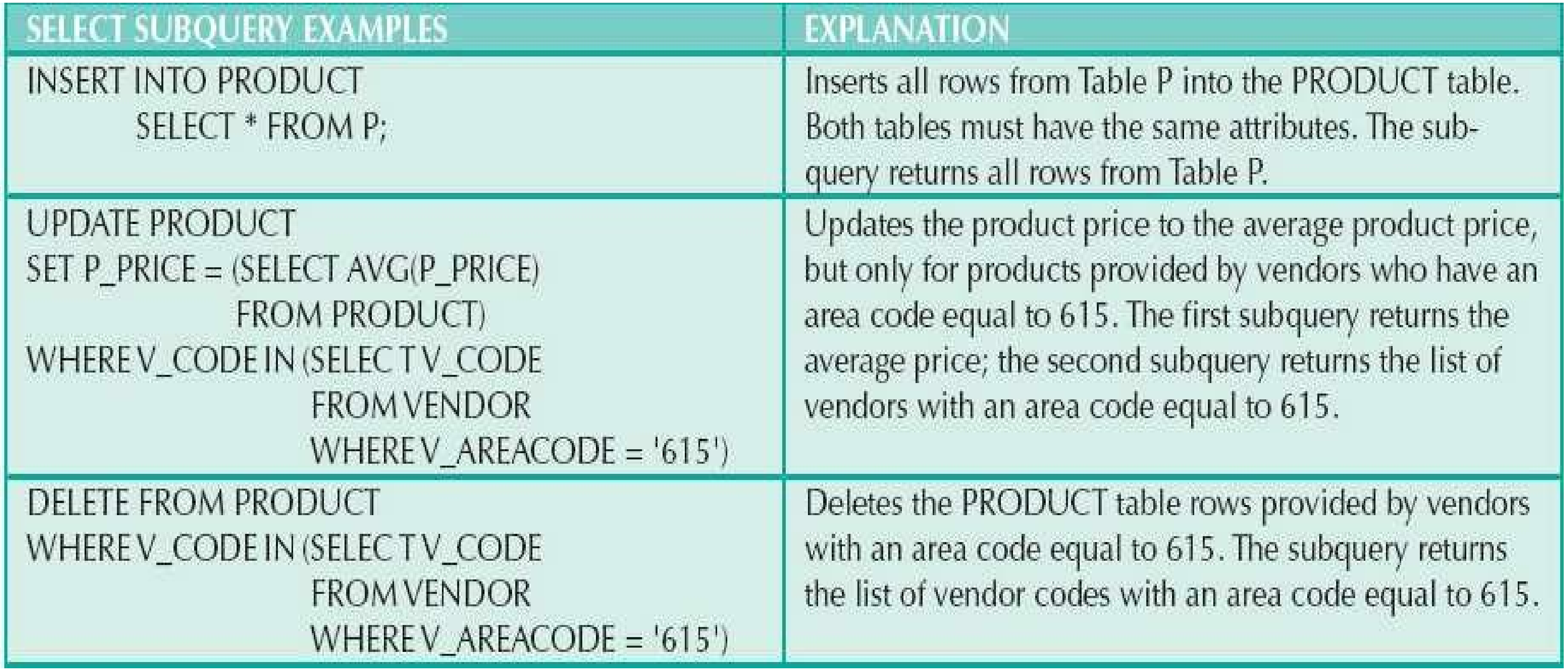
```
SELECT P_CODE, P_PRICE FROM PRODUCT
WHERE P_PRICE >= (SELECT AVG(P_PRICE) FROM PRODUCT);

SELECT DISTINCT CUS_CODE, CUS_LNAME, CUS_FNAME
FROM CUSTOMER JOIN INVOICE USING (CUS_CODE)
              JOIN LINE USING (INV_NUMBER)
              JOIN PRODUCT USING (P_CODE)
WHERE  P_CODE = (SELECT P_CODE FROM PRODUCT WHERE P_DESCRIPT = ‘Claw hammer’);

SELECT DISTINCT CUS_CODE, CUS_LNAME, CUS_FNAME
FROM CUSTOMER JOIN INVOICE USING (CUS_CODE)
           JOIN LINE USING (INV_NUMBER)
           JOIN PRODUCT USING (P_CODE)
WHERE      P_DESCRIPT = ‘Claw hammer’;
```

Task: Write subqueries.

Writing Subqueries



SELECT SUBQUERY EXAMPLES	EXPLANATION
INSERT INTO PRODUCT SELECT * FROM P;	Inserts all rows from Table P into the PRODUCT table. Both tables must have the same attributes. The subquery returns all rows from Table P.
UPDATE PRODUCT SET P_PRICE = (SELECT AVG(P_PRICE) FROM PRODUCT) WHERE V_CODE IN (SELECT V_CODE FROM VENDOR WHERE V_AREACODE = '615')	Updates the product price to the average product price, but only for products provided by vendors who have an area code equal to 615. The first subquery returns the average price; the second subquery returns the list of vendors with an area code equal to 615.
DELETE FROM PRODUCT WHERE V_CODE IN (SELECT V_CODE FROM VENDOR WHERE V_AREACODE = '615')	Deletes the PRODUCT table rows provided by vendors with an area code equal to 615. The subquery returns the list of vendor codes with an area code equal to 615.

Task: write subqueries

USIU AFRICA

Referential integrity using SQL.

RECAL

Referential integrity means that, if the foreign key contains a value, that value must refer to an existing, valid row in the parent table.

PRIMARY KEY (V_CODE)
In PRODUCT table the CREATE TABLE sequence, enforces referential integrity by specifying the following in the PRODUCT table:
FOREIGN KEY (V_CODE) REFERENCES VENDOR ON UPDATE CASCADE

ANSI SQL permits the use of ON DELETE and ON UPDATE clauses to cover CASCADE, SET NULL, or SET DEFAULT.

Task: As at least 5 different examples of supertypes and subtypes.

USIU AFRICA

Database triggers and stored procedures.

- A trigger is procedural SQL code that is automatically invoked by the RDBMS upon the occurrence of a given data manipulation event.
- A trigger is:
    - Called before or after a data row is inserted, updated, or deleted.
    - Associated with a database table.
    - Executed as part of the transaction that triggered it.
    - Present, optionally, in one or more times in a database table.

Task: discuss triggers and how they are relevant

Database triggers and stored procedures.

Uses of triggers

- Help in proper database operation and management, eg. to enforce constraints that cannot be enforced at the DBMS design and implementation levels.
- Automate critical actions and providing appropriate warnings and suggestions for remedial action eg. facilitating the enforcement of referential integrity.
- Update table values, insert records in tables, and call other stored procedures.
- Auditing purposes (creating audit logs)-Oracle
- Automatic generation of derived column values
- Enforcement of business or security constraints
- Creation of replica tables for backup purposes

Task: discuss triggers and how they are relevant

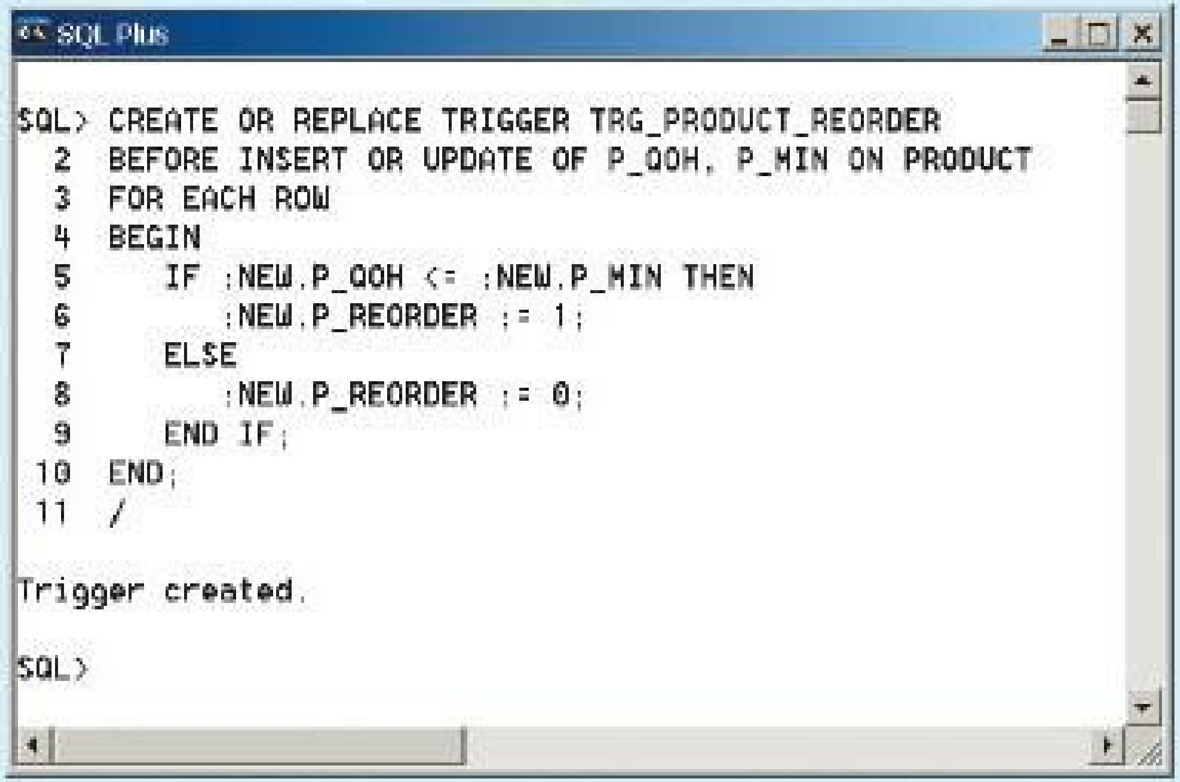
Database triggers and stored procedures.

Declaration-Oracle

```
CREATE OR REPLACE TRIGGER trigger_name
[BEFORE / AFTER] [DELETE / INSERT / UPDATE OF column_name] ON table_name
[FOR EACH ROW]
[DECLARE]
[variable_namedata type[:=initial_value] ]
BEGIN
PL/SQL instructions;
.........
END;
```

Task: discuss triggers and how they are relevant

Database triggers and stored procedures.



```
SQL> CREATE OR REPLACE TRIGGER TRG_PRODUCT_REORDER
     BEFORE INSERT OR UPDATE OF P_QOH, P_MIN ON PRODUCT
     FOR EACH ROW
     BEGIN
        IF :NEW.P_QOH <= :NEW.P_MIN THEN
           :NEW.P_REORDER := 1;
        ELSE
           :NEW.P_REORDER := 0;
        END IF;
     END;
     /

Trigger created.

SQL>
```

Task: discuss triggers and how they are relevant

Database triggers and stored procedures.

Stored procedure

- **Is a named collection of SQL statements.**

They are also stored in the database.

Advantages

- **They can be used to encapsulate and represent complete business transactions.**
- **They reduce network traffic and increases system performance.**
- **They also help reduce code duplication by creating unique PL/SQL modules that are called by the application programs.**
- **They minimizing the chance of errors and the cost of application development and maintenance.**

Task: discuss stored procedures and their relevance

USIU AFRICA

Database triggers and stored procedures.

Stored procedure

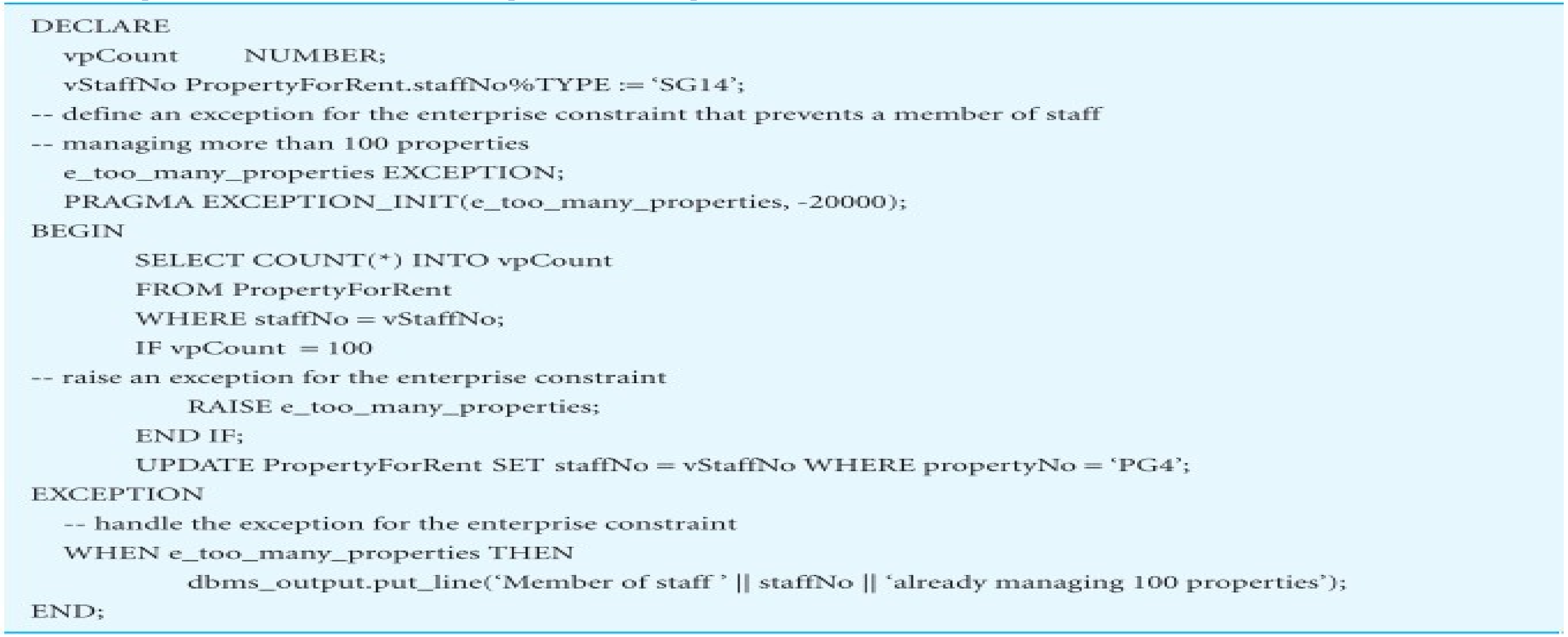
To create a stored procedure, you use the following syntax:

```
CREATE OR REPLACE PROCEDURE procedure_name [(argument [IN/OUT] data-type, … )]
            [IS/AS]
            [variable_namedata type[:=initial_value] ]
BEGIN
   PL/SQL or SQL statements;
   ...
END;
```

Task: discuss stored procedures and their relevance

Database triggers and stored procedures.

Stored procedure- example-exception handler

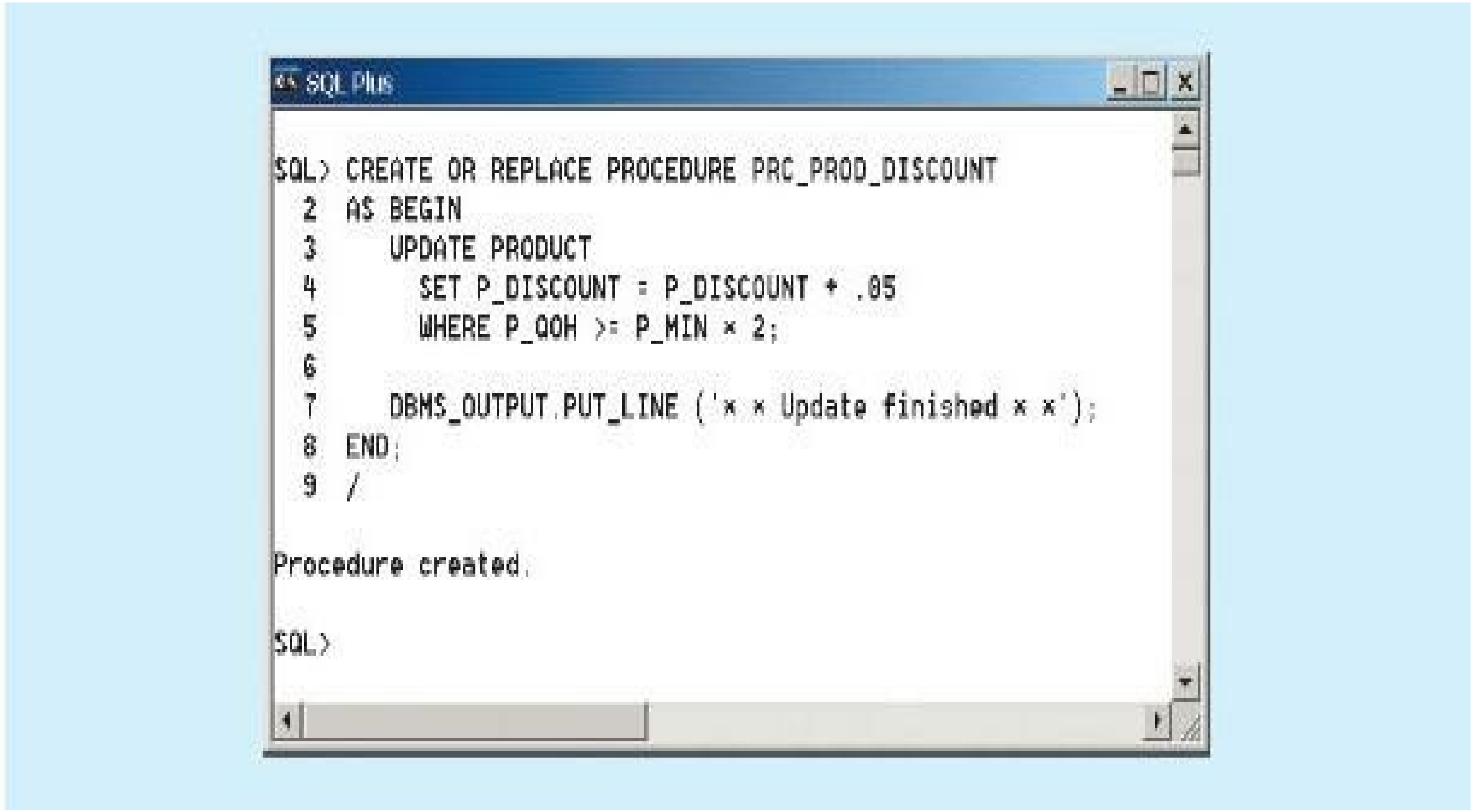


```
DECLARE
   vpCount       NUMBER;
   vStaffNo PropertyForRent.staffNo%TYPE := 'SG14';
-- define an exception for the enterprise constraint that prevents a member of staff
-- managing more than 100 properties
   e_too_many_properties EXCEPTION;
   PRAGMA EXCEPTION_INIT(e_too_many_properties, -20000);
BEGIN
          SELECT COUNT(*) INTO vpCount
          FROM PropertyForRent
          WHERE staffNo = vStaffNo;
          IF vpCount  = 100
-- raise an exception for the enterprise constraint
               RAISE e_too_many_properties;
          END IF;
          UPDATE PropertyForRent SET staffNo = vStaffNo WHERE propertyNo = 'PG4';
EXCEPTION
   -- handle the exception for the enterprise constraint
   WHEN e_too_many_properties THEN
               dbms_output.put_line('Member of staff ' || staffNo || 'already managing 100 properties');
END;
```

Task: discuss stored procedures and their relevance

Database triggers and stored procedures.

Stored procedure



```
SQL> CREATE OR REPLACE PROCEDURE PRC_PROD_DISCOUNT
  AS BEGIN
     UPDATE PRODUCT
       SET P_DISCOUNT = P_DISCOUNT + .05
       WHERE P_QOH >= P_MIN * 2;

     DBMS_OUTPUT.PUT_LINE ('* * Update finished * *');
  END;
  /

Procedure created.

SQL>
```

Task: **discuss stored procedures and their relevance**

Week 8 exercises

1) Write single-and multiple-table queries using SQL commands for EMPLOYEE table.
2) Define three types of join commands and use SQL to write these commands.
3) Write subqueries for the EMPLOYEE and PROJECT tables.
4) Implement referential integrity using SQL for the EMPLOYEE and PROJECT tables.
5) Describe common uses of database triggers and stored procedures and implement some for the VENDOR and PRODUCT tables.

Week 8 Session References

- **[Course Text] Carlos Coronel, Steven Morris, Peter Rob and Keeley Crockett Database Principles: Fundamentals of Design, Implementation, and Management, 14th Edition, 2022, ISBN-13978-0357673034.**

- **Thomas M. Connolly, Carolyn E. Begg (2021). Database Systems: A Practical Approach to Design, Implementation, and Management. Published by Pearson (July 14th 2021).**

Thank You