2

CODE SQL
STATEMENTS WITHIN PL/SQL

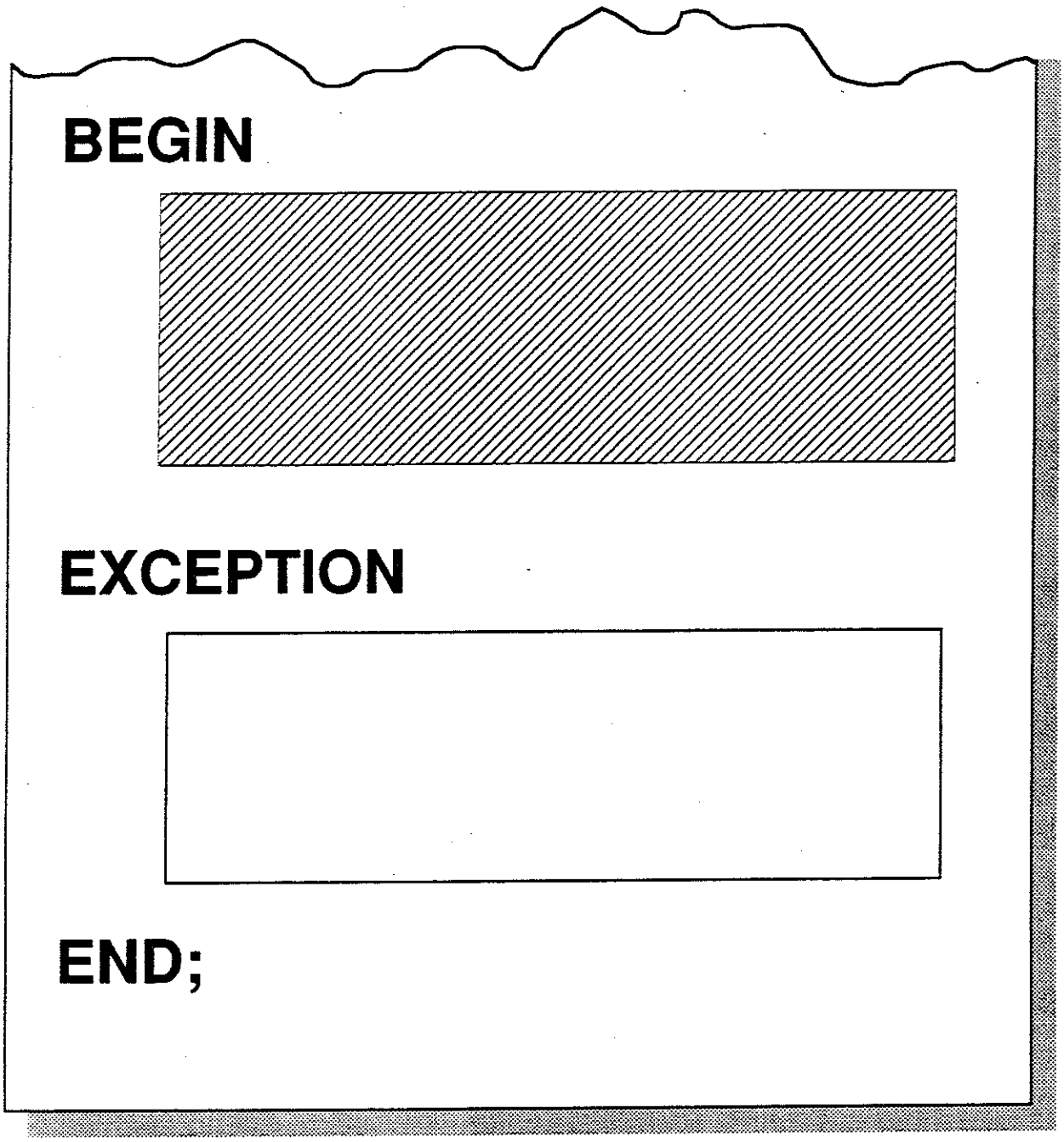
SECTION OBJECTIVES

At the end of this section, you should be able to:

1 Use SQL statements within PL/SQL procedures.

2 Identify SQL functions supported by PL/SQL.

CODE SQL STATEMENTS WITHIN PL/SQL: OVERVIEW



Code SQL Statements Within PL/SQL: Overview—cont'd

Write SQL Data Manipulation Language (DML) statements within a PL/SQL block.

SQL Statement Support

- INSERT
- UPDATE
- DELETE
- SELECT...INTO

Quick Notes

- The full ORACLE syntax is supported for these statements.
- A PL/SQL variable can be placed anywhere a constant may be legally placed.
- An identifier is first checked to see if it is a column in the database. If not, it is assumed to be a PL/SQL identifier.
- These statements may not appear as part of an expression.
- SQL DDL and DCL are not supported

CODE DML STATEMENTS

Example - INSERT

```
DECLARE
my_sal          NUMBER(7,2)     :=    3040.55;
my_ename        CHAR(25)        :=    'WANDA';
my_hiredate     DATE            :=    '08-Sep-88';

BEGIN
INSERT INTO emp (empno,ename,job,hiredate,sal,deptno)
VALUES (2741,my_ename,'CHAUFFEUR',my_hiredate,
          my_sal,20);
END;
```

Example - UPDATE

```
DECLARE
max_allowed     CONSTANT NUMBER := 5000;
good_cust       CHAR(8)         := 'VIP';

BEGIN
UPDATE accounts SET credit_limit = max_allowed
WHERE status = 'EMPLOYEE' OR status = good_cust;
END;
```

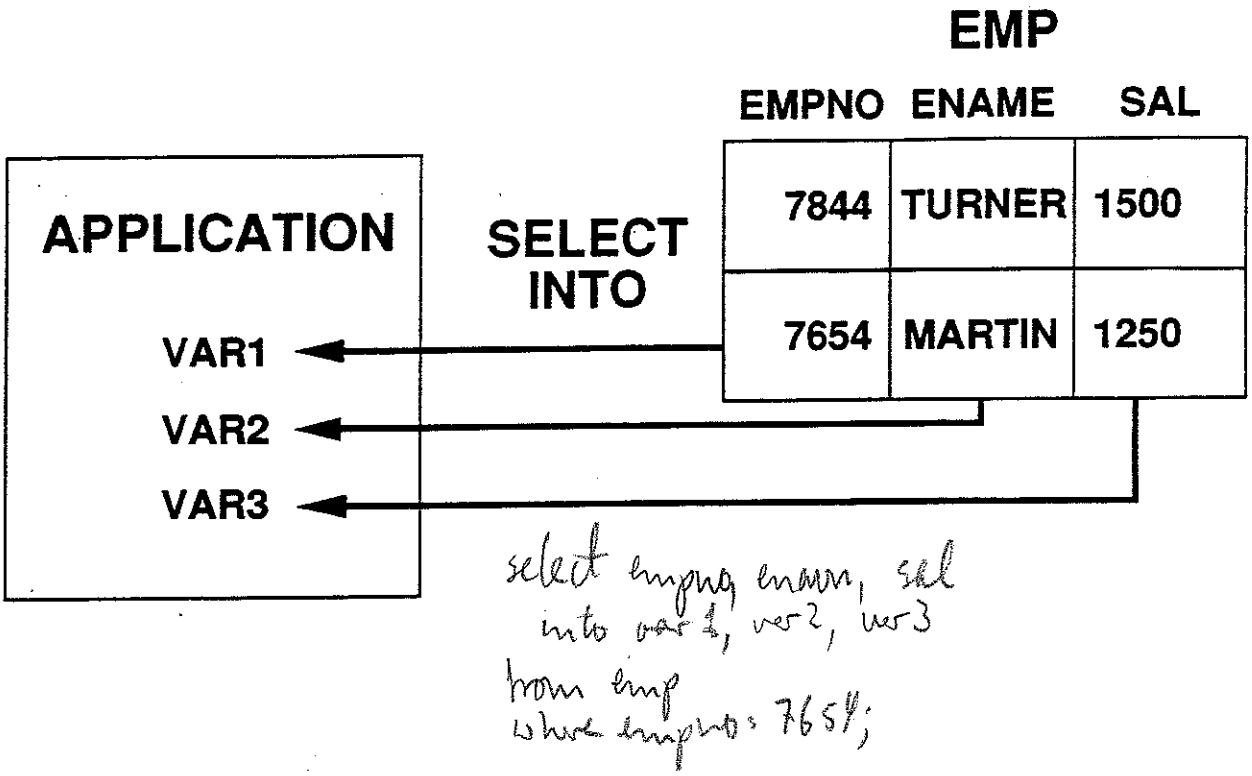
Example - DELETE

```
DECLARE
year_var      NUMBER := 1800;

BEGIN
DELETE FROM invention
     WHERE year < year_var;
END;
```

Code DML Statements—cont'd

SELECT INTO



Quick Notes

- A SELECT statement is the only DML that returns data. Provide a location for this data to be stored via the INTO clause.
- A SELECT...INTO statement must return exactly one row. Zero or multiple returned rows results in an error.
- For multi-row SELECTs use cursors.

Code DML Statements—cont'd

Syntax - SELECT

```
SELECT col1, col2,... INTO var1, var2 ...
     FROM table_name WHERE ...
```

Example - SELECT

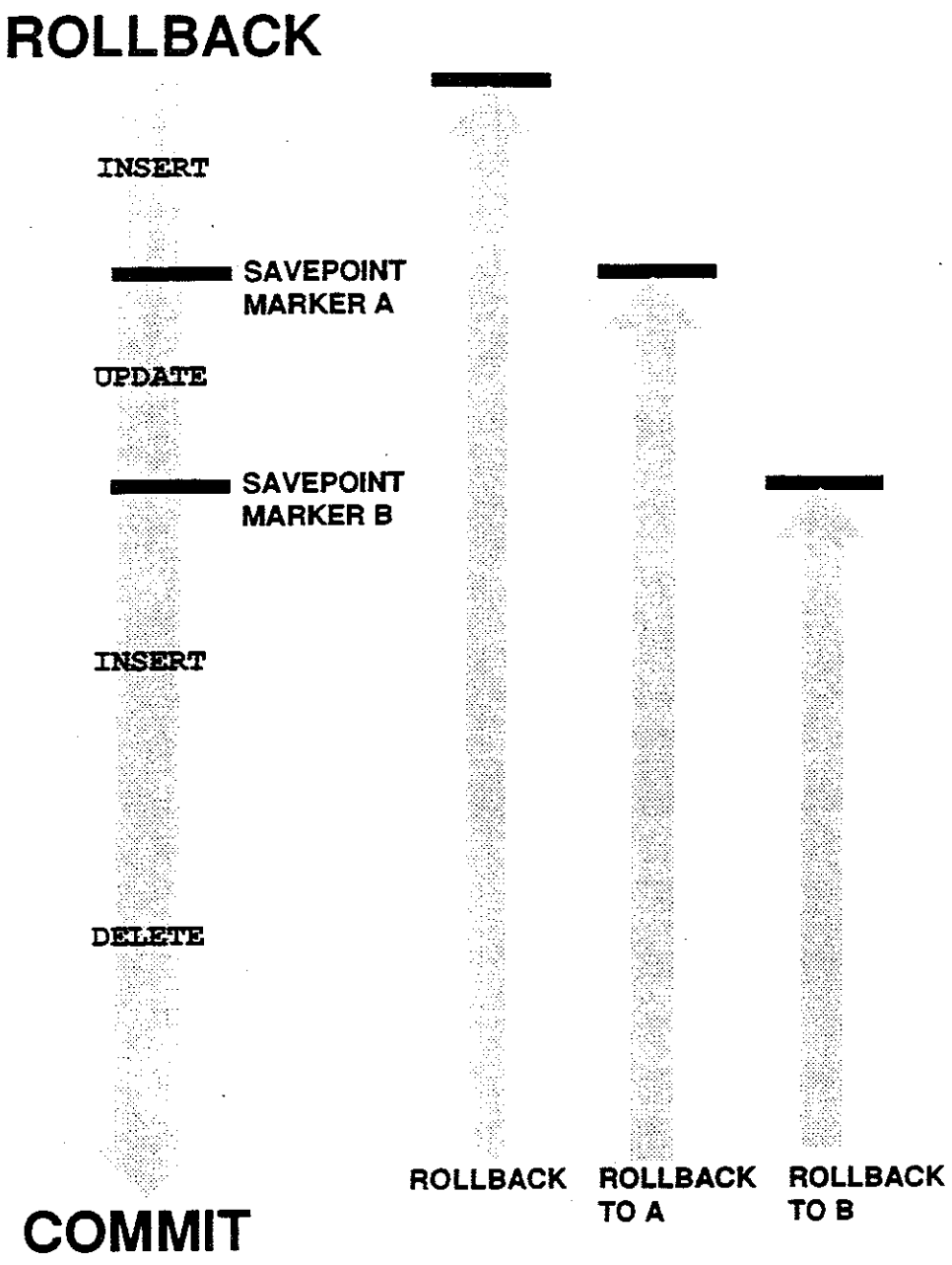
```
DECLARE
part_name          parts.name%TYPE;
num_in_stock       parts.num%TYPE;

BEGIN
SELECT name, num INTO part_name, num_in_stock
FROM parts WHERE part_id = 624;

          --manipulate the retrieved data here_

END;
```

PROCESS TRANSACTIONS



Process Transactions—cont'd

Syntax - SAVEPOINT

```
SAVEPOINT < marker_name >;
```

Syntax - ROLLBACK TO

```
ROLLBACK [WORK] TO [SAVEPOINT] < marker_name >;
```

Examples - SAVEPOINT and ROLLBACK TO

```
BEGIN
INSERT INTO temp (num_col1, num_col2, char_col)
        VALUES (1, 1, 'ROW 1');
SAVEPOINT A;
INSERT INTO temp (num_col1, num_col2, char_col)
        VALUES (2, 2, 'ROW 2');
SAVEPOINT B;
INSERT INTO temp (num_col1, num_col2, char_col)
        VALUES (3, 3, 'ROW 3');
SAVEPOINT C;
  ...
ROLLBACK TO SAVEPOINT B;
COMMIT;
END;
```

sql % rowcount => inneholder count fra siste
sql setning.

```
set serveroutput on;
declare
  :
Begin
  SQL stm...
  If SQL rowcount < 10 then
    commit;
    DBMS_PUT.LINE ('Jeg committet');
  ELSE
    rollback;
    DBMS_PUT_LINE ('Har rullet tilbake');
  END IF;
END;
/
```

Skal
skal være
DBMS.output.put_line ('')

for at &&val skal spørres etter må man ha set scan ON

REFERENCE BUILT-IN FUNCTIONS

Reference built-in functions in SQL statements and procedural statements.

Reference-able Functions in SQL Statements

- Numeric (for example, SQRT, ROUND, POWER)
- Character (for example, LENGTH, UPPER, SUBSTR,)
- Date (for example, ADD_MONTHS, MONTHS_BETWEEN)
- Group (for example, AVG, MAX, COUNT)
- Conversion (for example, TO_CHAR, TO_DATE, TO_NUMBER)
- Miscellaneous (for example, LEAST, NVL, DECODE)

Example - Use functions in a DML statement

```
INSERT INTO phonebook (lastname) VALUES
      (UPPER(my_lastname));
```

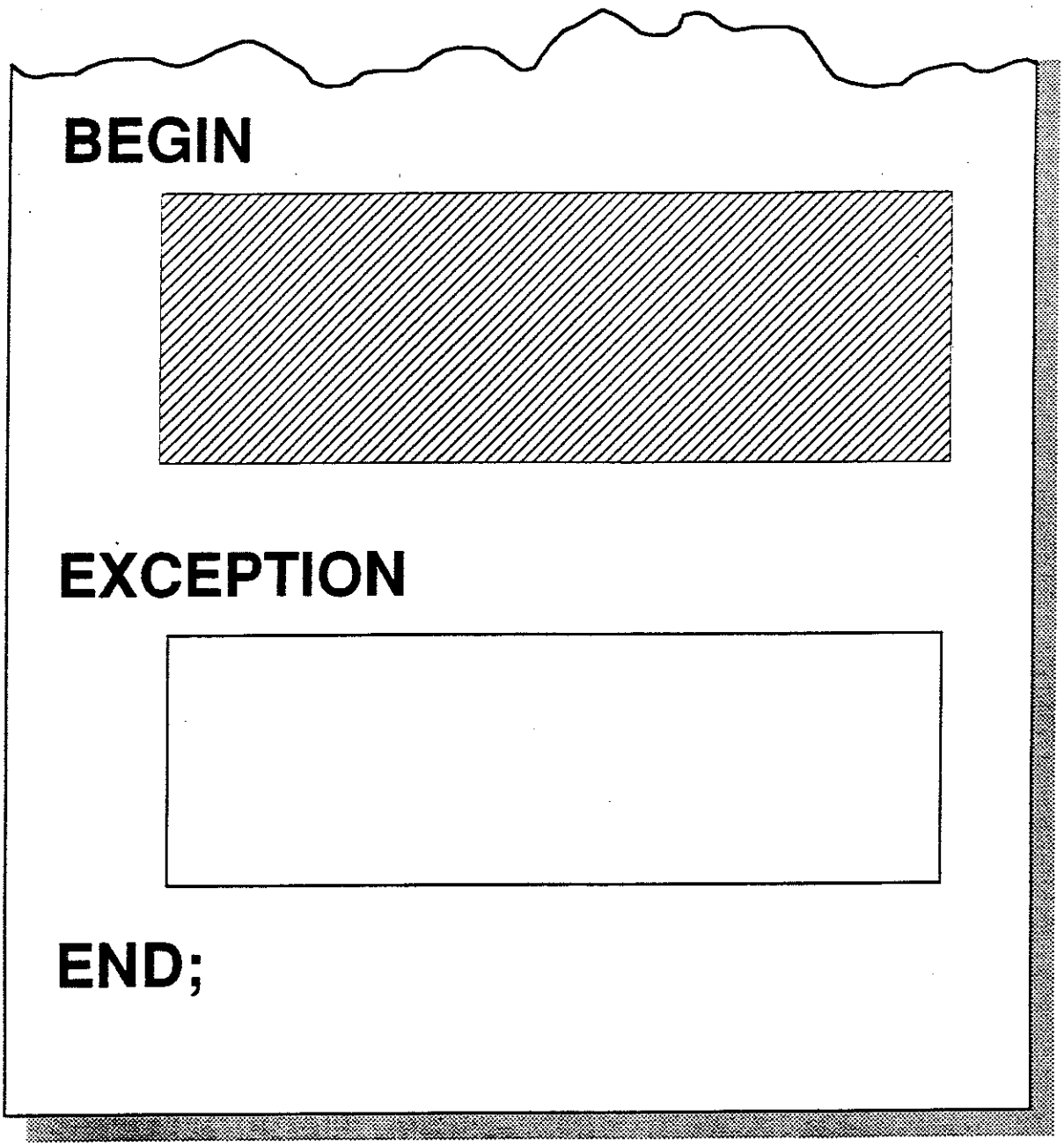
Reference-able Functions in Procedural Statements

- SQL functions listed above except for DECODE and Group functions
- Error reporting (for example, SQLCODE, SQLERRM)

Example - Use functions in an assignment statement

```
x := SQRT(y);
lastname := UPPER(lastname);
age_diff := MONTHS_BETWEEN(birthday1, birthday2)/12;
```

USE SQL WITHIN PL/SQL: SUMMARY



SQL Statement Support

- INSERT
- UPDATE
- DELETE
- SELECT...INTO