

--PL/SQL

Format for an anonymous block

```
DECLARE
    DATANAME DATATYPE [:= value]

    BEGIN
        code
    EXCEPTIONS

END;
/
```

Format for a procedure

```
CREATE OR REPLACE PROCEDURE PROCEDURENAME ( )
IS/AS
    DATANAME DATATYPE [:= value];

    BEGIN
        code
    EXCEPTIONS

END;
/
```

Format for a function

```
CREATE OR REPLACE FUNCTION FUNCTIONNAME ( ) RETURN DATATYPE
IS/AS
    DATANAME DATATYPE [:= value];

    BEGIN
        code
        RETURN VAR_OF_DATATYPE;
    EXCEPTIONS

END;
/
```

Format for a package

```
CREATE OR REPLACE PACKAGE PACKAGENAME
IS
    PROCEDURE PROCNAME ( );
    FUNCTION FUNCTIONNAME ( ) RETURN DATATYPE;
END;
```

Format for a package body

```
CREATE OR REPLACE PACKAGE BODY PACKAGENAME
IS
    PROCEDURE PROCNAME ( )
    IS
```

```

        code
END PROCNAME;

FUNCTION FUNCTIONNAME ( ) RETURN DATATYPE
IS
    code
END FUNCTIONNAME;
END;

```

Cursors

For 1 row - SELECT _ INTO

For more than one row - CURSOR CNAME IS
SELECT

The cursor goes through the ResultSet

Triggers

- Execute SQL statements during certain actions
- We will restrict triggers to "for each row" type triggers
- Can be done before or after an event, most of the time, before trigger will be used
- INSERT, UPDATE, and DELETE can be used to make changes
- create or replace trigger TRIGGERNAME
 - before update
 - on viewing
 - for each row
 - begin
 - :new.viewdate := sysdate;
 - end;
 - /
- show errors;
- commit;
- select * from viewing;
- update viewing set comment="New Comment" where clientno="CR56" and propertyno="PA14";
- commit;

- before update or insert - trigger is executed after both an update or insert
- row level triggers - default used
 - before or after
 - for each row
 - insert, update, or delete
- statement level triggers -
- instead of - trigger on a view
- schema
 - create, alter, or drop

HW 3 Hints

-An outside loop will traverse the column, while an inside loop will traverse constraints