Week 9 Tutoring

CSE 180

Stored Function

```
CREATE OR REPLACE FUNCTION
        function_name (<arguments> <type>)
RETURNS <type> AS $$


// Declare variables
DECLARE
        variables variable_type;
        ... more variables;
```

Cursors

DECLARE CURSOR FOR

SELECT <attributes>

FROM <table names>

<WHERE conditions> ;

Iterate Cursor

```
BEGIN
    <variables/ any conditions you want to check>
    OPEN <cursor name>
    LOOP
        FETCH <cursor name> INTO <attributes in cursor>
        <anything you want to do>
```

Finish Stored Function

```
        END LOOP ;  // close the loop and cursor

        CLOSE <cursor name>;

        RETURN <value>

END

$$ LANGUAGE plpgsql;
```

Queries

```
char query[MAXSQLSTATEMENTSTRINGSIZE]
PGresult *res;
sprintf(query, "<SQL Statement>”);
res = PQexec(conn, query);
```

Query Results

```
PQgetvalue(res, 0, 0); // get the first row, first element
PQgetvalue(res, 0, 1); // get the first row, second element
```

Counting

Tuples in the result

```
int n_tuples = PQntuples(res);
```

Tuples that were updated

```
int count = atoi(PQcmdTuples(res));
```

or

```
GET DIAGNOSTICS integer_var = ROW_COUNT;
```

Statuses

PGRES_TUPLES_OK	Indicates successful completion of a SELECT
PGRES_COMMAND_OK	Indicates successful completion of a command that does not return rows

```
if (PQresultStatus(res) != PGRES_TUPLES_OK) {
        fprintf(stderr, "Query failed: %s", PQerrorMessage(conn));
        PQclear(res);
        bad_exit(conn);
}
```