Appendix III

**Object Privileges**

Access to operations privileges.

ORACLE in general is regulated via 'user privileges'. permitted on individual tables is regulated through

The type of object-related

**Object-related Privileges**

Every database object (eg. table, view. index, etc.) has an owner. In general, owners have full object privileges on their own objects, and they may also grant the same privileges to other users.

Privileges are granted using the GRANT command, and can be withdrawn using the REVOKE command.

**GRANT** {privilege,.. | ALL} **ON** object **TO** {user | PUBLIC } **REVOKE** {privilege,.. | ALL} **ON** object **FROM** {user | PUBLIC }

Object privileges for a table include:- SELECT, INSERT, UPDATE, DELETE

So, for example, if user ORA5432 owns a table called EMP1 and issues the command . . .

**GRANT** **SELECT,** **INSERT** **ON** EMP1 **TO** ORA1234

. . . then user ORA1234 will be able to read information from table EMP1, and/or insert new rows (but will not be able to update rows, or delete rows).

*(To* *use* *this* *table,* *ORA1234* *will* *have* *refer* *to* *the* *table* *EMP1* *as* *ORA5432.EMP1* *in* *the* *FROM* *statement* *of* *any* *query.)*

And, . . .

**GRANT** **ALL** **ON** EMP1 **TO** **PUBLIC**

. . . would grant ALL privileges on EMP1 to all users (all users have the PUBLIC role)

To reverse the granting of these privileges, use REVOKE, as in:

**REVOKE** **SELECT,** **INSERT** **ON** EMP1 **FROM** ORA1234

**REVOKE** **ALL** **ON** EMP1 **FROM** **PUBLIC**

The use of **GRANT** and **REVOKE** facilitates a team-based approach to database development.

SQL Workbook 73 September 2020

Appendix III

SQL Workbook 74 September 2020