Chapter 1 and 2

A database consists of an organized collection of data. Various models: **Hierarchical** (tree structure), **Network** (allows more than one parent per child), **Relational** , Object Relational, Object Oriented Relational **Database Management System** (DBMS) is a set of computer programs that controls the creation, maintenance, and the use of a database.
A **database administrator** (DBA) is a person responsible for the implementation, maintenance, and repair of an organization's database. A **data analyst** is a person responsible for analyzing data requirements within an organization and modeling the data.
SQL is referred to as **Structure Query Language**. It is a database computer language designed for managing data in relational database management systems (RDBMS).
Data Definition Language (DDL) statements are used to define the database structure or schema. Examples include: Create, Alter, Drop, Truncate, Comment, Rename. **Data Manipulation Language** (DML) statements are used for managing data within schema objects. Examples include: Select, Insert, Update, Delete, Merge, Call, Explain Plan, Lock Table. **Data Control Language** (DCL) statements. Examples include: Grant, Revoke
Transaction Control (TCL) statements are used to manage the changes made by DML statements. It allows statements to be grouped together into logical transactions. Examples include: Commit, Savepoint, Rollback, Set Transaction

Normalization is the process of refining tables, keys, columns, and relationships to create an efficient database.
Normal Forms: **First Normal Form** (each row and column contain one and only one value, must be atomic), **Second Normal Form** (every non-key column MUST depend on the entire primary key), **Third Normal Form** (No non-key column depends on another non-key column), **Fourth Normal Form** (Forbids independent relationships between primary key and non-primary key columns), **Fifth Normal Form** (Breaks tables into the smallest possible pieces in order to remove redundancy).
Primary Key is a field whose values are unique for each record in a table. **Unique/Candidate/Alternate Key** is a combination of attributes that can be uniquely used to identify a database record without any extraneous data. Can have several of these per table.
Foreign Key is a key field that identifies records in a different table.
Entity Relationship Diagrams (ER) is a conceptual model that shows the structural organization of entities, relationships, and attributes. One-to-one, One-to-many, Many-to-many

Chapter 3

VARCHAR2 (n): Variable-length character data where n represents the column's maximum length. Max is 4000 characters.
CHAR (n): Fixed-length character columns where n represents the column's length. The default is 1 and the maximum is 2000.
Number (p, s): Numeric column where p indicates precision and s indicates scale.
DATE: Stores date and time between January 1, 4712 BC and December 31, 9999 AD. Oracle's default date format is DD_MON-YY.
DESC Command provides information about the columns in a table using DESC table_name;
View contents of table: SELECT * FROM table_name;
Drop table command: Drop TABLE table_name;
Flashback (a dropped table can be recovered): FLASHBACK TABLE table_name TO BEFORE DROP;
Recyclebin (purging from the recyclebin): PURGE RECYCLEBIN;
NOT NULL and DEFAULT constraints: NUMBER NOT NULL, CHAR DEFAULT 'm'
Insert Statement: INSERT INTO table_name VALUES (…, …, …);
View Contents: SELECT * FROM table_name;
Data Dictionary Tables: DESC user_tables;
Delete contents inside of a table (recoverable): DELETE FROM table_name;
Truncate (non-recoverable): TRUNCATE TABLE table_name;
Renaming a table: RENAME table_name TO another_name;
Additional Column: ALTER TABLE table_name ADD column_name NUMBER;
Adding multiple columns: ALTER TABLE table_name ADD (Height NUMBER, Weight NUMBER);
Modify: ALTER TABLE table_name MODIFY column_name TYPE;

- Cannot change between data types, VARCHAR cannot be less than existing data

Dropping a column (cannot be changed): ALTER TABLE table_name DROP (column_name);
Set Unused: ALTER TABLE table_name SET UNUSED (column_name1, column_name2);

- ALTER TABLE table_name DROP UNUSED COLUMNS;

Renaming column (one): ALTER TABLE table_name RENAME COLUMN colum_name TO another_name;
Primary Key at Column Level: column_name TYPE PRIMARY KEY

- Primary Key at Table Level: CONSTRAINT constraint_name_pk PRIMARY KEY (column_name)
- Composite Primary Key: CONSTRAINT constraint_name_pk PRIMARY KEY (column1, column2);

- Using ALTER TABLE: ALTER TABLE table_name ADD PRIMARY KEY (column_name);
- ALTER TABLE table_name MODIFY column_name PRIMARY KEY;
- ALTER TABLE table_name ADD PRIMARY KEY (column1, column2);

Unique at column level (can have several of these unique PK): column_name TYPE UNIQUE;

- Table level: CONSTRAINT table_name_uk UNIQUE (column_name_uk);
- Using alter command: ALTER TABLE table_name ADD CONSTRAINT constraint_name_uk UNIQUE (column_name);
- ALTER TABLE table_name MODIFY column_name UNIQUE;

Check constraint: column_name TYPE CHECK (comparison)

- Using alter command: ALTER TABLE table_name MODIFY column CONSTRAINT constraint_name_ck CHECK (comparison);
- ALTER TABLE table_name ADD CHECK (comparison);
- Between Clause: column_name TYPE CONSTRAINT constraint_name_ck CHECK (comparison BETWEEN 10 and 100);
- Char constraint: column_name TYPE CHECK (column_name in ('…', '…'))

Not NULL constraint: column_name TYPE NOT NULL

- Using modify command: ALTER TABLE table_name ADD column NOT NULL;

References Keyword: column_name TYPE CONSTRAINT table_name_fk REFERENCES table

- Column_name TYPE REFERENCES table_name (column_name)
- Notice that the parent tables have to be populated first before any child records are inserted.

Foreign Keys: FOREIGN KEY (column_name ON THIS TABLE) REFERENCES table_name ON DELETE CASCADE

- Delete Cascade: column_name TYPE REFERENCES table_name ON DELETE CASCADE
- Using alter command: ALTER TABLE table_name ADD FOREIGN KEY (column_name) REFERENCES other_table;

Disable Constraints: ALTER TABLE table_name DISABLE CONSTRAINT constraint_name;

Enable Constraints: ALTER TABLE table_name ENABLE CONSTRAINT constraint_name;

Drop Constraints: ALTER TABLE table_name DROP CONSTRAINT constraint_name;

Primary Key with Cascade Options: ALTER TABLE table_name DROP PRIMARY KEY CASCADE;

Indexes: CREATE INDEX index_name ON table_name (column_name);

- SELECT index_name FROM user_indexes WHERE table_name='TABLENAME';
- Alter index command: ALTER INDEX index_name RENATE TO different_name;
- Dropping an index: DROP INDEX index_name;

User_constraints: SELECT table_name, constraint_name, constraint_type FROM user_constraints WHERE table_name='TABLENAME';

YOU CANNOT USE TRUNCATE ON A TABLE REFERENCED BY A FOREIGN KEY CONSTRAINT, INSTEADE USE DELETE STATEMENT.
WHEN GETTING RID OF THE TABLES, THE CHILD TABLE HAS TO BE DELETE PRIOR TO THE PARENTS.

Chapter 4

Inserting Textual Data: INSERT INTO table_name (cola, colb, colc) VALUES ('…', '…', '…');

Inserting Numerical Data: INSERT INTO table_name (column_name) VALUES (….);

- The NUMBER data type can store real or whole numbers. NUMBER (5) will store whole numbers that don't exceed 5 digits and it will round real numbers. NUMBER (5, 3) can store a real number with two digits as a whole number and 3 digits after the decimal place.

Dates: Month (April), MON (APR), MM (04), RM (IV), D (4 ⍰ Indicates Wednesday), DD (28), DDD (365), DAY (WEDNESDAY), DY (WED), YYYY (2009), YYY or YY or Y (009, 09, 9), YEAR (TWO THOUSAND NINE), BC or AD (2009 A.D.)

Time: SS (0-59), SSSS (0-86399), MI (0-59), HH or HH12 (12), HH24 (0-23), A.M. or P.M. (A.M. or P.M.)

Inserting Dates: INSERT INTO table_name VALUES ('01 feb 11');

- Display in different format: SELECT TO_CHAR (column_name, 'YYYY/MM/DD hh24:mi:ss') FROM table_name;
- Inserting in different format: INSERT INTO table_name VALUES (TO_DATE ('99/02/04 23:23:20', 'YY/DD/MM HH24:MI:SS));
- Alter Session format: ALTER SESSION SET nls_date_format = 'MON-DD-YY HH24:MI:SS';

Sequences: CREATE SEQUENCE table_name_seq INCREMENT BY 4 START WITH 10;

- Get access to next sequences: INSERT INTO table_name VALUES (table_name_seq.nextval);
- Get access to current sequence: SELECT table_name_seq.currval FROM DUAL;
- User sequences table: SELECT sequence_name FROM user_sequences;
- Alter sequences: ALTER SEQENCE table_name_seq INCREMENT BY 2;
- Drop Sequences: DROP SEQUENCE sequence_name;