**TERMINOLOGY**

First Normal Form: Each row and column intersection must contain one and only one value. Must be atomic. Eliminate repeating groups.

Second Normal Form: Every non-key column must depend on the entire primary key. Eliminate partial key dependencies.

Third Normal Form: No non-key column depends on another non-key column. Eliminate transitive dependencies.

Fourth Normal Form: Forbids independent relationships between primary key columns and non-key columns.

Fifth Normal Form: Break tables into the smallest possible pieces in order to eliminate redundancy.

Atomic: One basic unit. Cannot be broken down into anything smaller.  
Composite Key: A key made up of multiple columns. May be foreign key. May be used as a primary key.

PRIMARY KEY: Must be unique. Must not be null. May be composed of one or multiple columns (composite key).

Unique/Candidate/Alternate Key: Each table may have one or more candidate keys. One of these candidate keys is selected as the primary key. Alternative key used strictly for data retrieval purposes. May be a composite key.

FOREIGN KEY: A key field that identifies records from a different table. Usually another table’s primary key. In 1-to-Many relationships the foreign key goes on the “Many” side. It must exist in the “one” side or it isn’t added. Child records must be deleted before parent records are deleted. Cannot use TRUNCATE on table.

UNIQUE: Ensures all data values stored in a specific column are unique. It allows NULL values, unlike Primary Key.

CHECK: Ensures that specified condition is true before the data is added. For example death date > birth date.

NOT NULL: Ensure a column can’t contain a NULL value.

One-to-one: One instance from table A is associated with one instance from table B.

One-to-many: One instance from table A is associated with many instances from table B.

Many-to-many: Many instances from table A are associated with many instances from table B. This relationship requires a join table. The join table contains a composite primary key utilizing foreign keys from both tables.

**DATATYPES**

VARCHAR2(**n**) Variable-length character data. **n** represents the column’s maximum length. **Max**: 4000 characters.

CHAR(**n**) Fixed-length character columns. **n** represents the column’s length. **Default**: 1 **Max**: 2000 characters.

NUMBER(**p**,**s**) Numeric column. **p** indicates precision (total number of digits). **s** indicates scale (decimal digits).

DATE Stores date and time between January 1, 4712 BC and December 31, 9999 AD. **Default** format: DD-MON-YY

OTHER DATATYPES: BINARY\_FLOAT, BINARY\_DOUBLE, INTEGER, LONG, CLOB, RAW, LONG RAW, BLOB, BFILE, TIMESTAMP, INTERVAL

**SYNTAX**

CREATE TABLE tablename(

columnName TYPE,

etc...

);

DESC tablename; Prints out table layout.

SELECT \* FROM tablename;

SELECT colname FROM tablename WHERE somecol=somevalue;

DROP TABLE tablename;

ALTER TABLE sometable DROP column;

FLASHBACK TABLE tablename; Brings table back from recycle bin

PURGE RECYCLEBIN; Purges recycle bin

**NOT NULL** and **DEFAULT EXAMPLE:**

CREATE TABLE gender(

gender\_id NUMBER NOT NULL,

gender CHAR DEFAULT ‘m’

);

SELECT \* FROM user\_tables; Lists all tables belonging to user

DELETE FROM tablename; Deletes all rows from table. Can be applied to all or only specific rows. (NO ASTERISK USED)

RENAME sometable TO newtablename; Renames a table.

**ADD COLUMN EXAMPLE: Columns will be added to the end of the table**

ALTER TABLE sometable ADD somecolumn NUMBER;

ALTER TABLE sometable ADD(

column1 VARCHAR2(10),

column2 NUMBER,

);

SET UNUSED COLUMNS: ALTER TABLE sometable SET UNUSED (column2, column3);

DROP UNUSED COLUMNS: ALTER TABLE sometable DROP UNUSED COLUMNS;

RENAME COLUMN: ALTER TABLE sometable RENAME COLUMN somecolumn TO anothercolumn;

**MODIFY EXAMPLE: Columns must not break modifications (old data has to fit new sizes)**

MODIFY COLUMN: ALTER TABLE sometable MODIFY column1 VARCHAR2(20);

**SPECIAL TABLES:**

SELECT table\_name, constraint\_name, constraint\_type FROM user\_constraints WHERE table\_name=’sometable’;

**CONSTRAINTS:**

**At COLUMN level use ‘CONSTRAINT’ command to name constraint**

CREATE TABLE sometable(

ID NUMBER CONSTRAINT sometable\_ID\_pk PRIMARY KEY,

name VARCHAR2(20) NOT NULL,

age NUMBER CONSTRAINT sometable\_age\_ck CHECK (age>0 AND age<100)

);

**COMPOSITE PRIMARY KEY (only possible at TABLE level)**

... CONSTRAINT sometable\_col1\_col2\_pk PRIMARY KEY (col1, col2) ...

ALTER TABLE sometable ADD PRIMARY KEY (col1, col2);

**UNIQUE CONSTRAINT (can be added to any column during table creation)**

... tableID NUMBER UNIQUE, …

... tableID NUMBER CONSTRAINT constraintname\_uk UNIQUE, …

… CONSTRAINT constraintname\_uk UNIQUE (tableID) …

… UNIQUE (tableID) …

ALTER TABLE sometable ADD CONSTRAINT constraintname\_uk UNIQUE (tableID);

**COMPOSITE UNIQUE (a combination of 2 or more columns must be unique)**

… CONSTRAINT constraintname\_uk UNIQUE (tableID, col1) …

ALTER TABLE sometable CONSTRAINT constraintname\_uk UNIQUE (tableID, col1);

**CHECK CONSTRAINT (can be added at COLUMN or TABLE level)**

... height NUMBER CHECK (height>5) … (exclusive)

... CHECK (height>5 AND height<10) … (exclusive)

… height NUMBER CHECK (height BETWEEN 5 AND 10) … (BETWEEN is inclusive)

ALTER TABLE sometable MODIFY height CONSTRAINT sometable\_height\_ck CHECK(height>10);

**NOT NULL CONSTRAINT (NOT NULL cannot be created at TABLE LEVEL)**

… height NUMBER NOT NULL …

ALTER TABLE sometable MODIFY height NOT NULL; (Cannot use ‘ADD’ keyword. Only ‘MODIFY’)

**FOREIGN KEY CONSTRAINT**

… foreignkey NUMBER REFERENCES anothertable …

... CONSTRAINT sometable\_foreignkey\_fk FOREIGN KEY (foreignkey) REFERENCES anothertable …

ALTER TABLE sometable ADD FOREIGN KEY (foreignkey) REFERENCES anothertable;

**DISABLE/ENABLE/DROP CONSTRAINTS**

**DISABLE** ALTER TABLE sometable DISABLE PRIMARY KEY; (doesn’t require column name)

ALTER TABLE sometable DISABLE UNIQUE (col1, col2);

**ENABLE** ALTER TABLE sometable ENABLE PRIMARY KEY; (doesn’t require column name)

ALTER TABLE sometable ENABLE CONSTRAINT sometable\_col1\_col2\_uk;

The status column in user\_constraints conveys if the constraint is enable or disabled.

**DROP** ALTER TABLE sometable DROP CONSTRAINT sometable\_height\_ck;

**DROP/DELETE/TRUNCATE**

**DROP**

DROP COLUMN(S): ALTER TABLE sometable DROP (column4, column5);

DROPPING TABLE: DROP TABLE sometable; (Cannot drop a table referenced by a foreign key without first dropping the child)

**TRUNCATE**

Cannot TRUNCATE a table which is referenced by a FOREIGN KEY CONSTRAINT. You must disable or remove the foreign key constraint from the child before truncating the parent. Use DELETE instead.

**DELETE CASCADE**

DELETE CASCADE … foreignkey1 NUMBER REFERENCES sometable ON DELETE CASCADE …

You can now delete records from the parent table without getting an error because the child keys are cascade deleted as well.

**INDEX (created automatically when a PRIMARY KEY is created)**

CREATE INDEX sometable\_column1\_idx ON sometable (column1);

Indexes are located in the user\_indexes table.

SELECT \* FROM user\_indexes WHERE table\_name=’sometable’;

**INSERTING DATA**

INSERT INTO sometable VALUES (‘data1’, ‘data2’, ‘data3’); (must be entered in the same order as the columns)

INSERT INTO sometable (col3, col2, col1) VALUES (‘data3’, ‘data2’, ‘data1’); (can use any order if listing the columns)

**NULL INSERT** INSERT INTO sometable VALUES (NULL, ‘col2’, ‘col3’); (no quotes around NULL)

NUMBER(5) (only accepts an integer and will round numbers on the first decimal)

NUMBER(5,2) (5 digits long total, 2 of which are decimal digits. Will round extra decimals.)

NUMBER(5,3) (won’t allow ‘123’ because only 2 digits are allowed to the left of the decimal)

**DATE**

DEFAULT FORMAT: DD-MON-YY 19-APR-99

Month Month spelled out

MON 3-letter abbreviation

MM 2-digit month

RM Roman Numeral month

D numerical day of the week

DD numerical day of the month

DDD numerical day of the year

DAY named day of the week

DY 3-letter abbreviation for day

YYYY 4-digit year

YYY/YY/Y numerical value for last 3, 2 or 1 digit of the year

YEAR spelled out version of the year

BC/AD indicates BC or AD

SS seconds

SSSS seconds past midnight

MI minutes

HH/HH12 hours

HH24 military time

AM/PM indicates AM or PM

INSERT INTO sometable (date\_col) VALUES (SYSDATE); Inserts current system date/time.

INSERT INTO sometable (date\_col) VALUES (TO\_DATE(‘99/02/15’, ‘YY/MM/DD’)); Using TO\_DATE function.

SELECT TO\_CHAR (date\_col, ‘YY/MM/DD’) FROM sometable; Outputs using TO\_CHAR function.

**SEQUENCES**

CREATE SEQUENCE sometable\_tableID\_seq START WITH 1; Creates a sequence that starts with 1

CREATE SEQUENCE sometable\_tableID\_seq INCREMENTBY 4 START WITH 10; Starts with 10 and increments by 4

**(use ‘nextval’ to get the next value in the sequence)**

INSERT INTO TABLE sometable (tableID, name, age) VALUES (sometable\_tableID\_seq.nextval, ‘John’, 23);

**(use ‘currval’ to get the last used value in the sequence)**

SELECT \* FROM sometable WHERE (tableID=sometable\_tableID\_seq.currval); Shows the last entered record using the currval

Sequences are stored in the user\_sequences table.

SELECT sequence\_name FROM user\_sequences; Will list the sequences in the table by name.

DROP SEQUENCE sometable\_tableID\_seq;