RELATIONAL DATABASE STRUCTURE

Definition of Terms

Relation: A relation is a table with columns and rows.

Attribute: An attribute is a named column of a relation.

Domain: A domain is the set of allowable values for one or more attributes.

Tuple: A tuple is a row of a relation.

Degree: The degree of a relation is the number of attributes it contains.

Cardinality: The cardinality of a relation is the number of tuples it contains.

Properties of Relations

A relation has the following properties:

- The relation has a name that is distinct from all other relation names in the relational schema;
- Each cell of the relation contains exactly one atomic (single) value;
- Each attribute has a distinct name;
- The values of an attribute are all from the same domain;
- Each tuple is distinct; there are no duplicate tuples;
- The order of attributes has no significance;
- The order of tuples has no significance.

Integrity Constraints

i) Entity Integrity

Entity integrity applies to the primary keys of base relations.

In a base relation, no attribute of a primary key can be null.

ii) Referential Integrity

Referential integrity applies to foreign keys.

If a foreign key exists in a relation, either the foreign key value must match a candidate key value of some tuple in its home relation or the foreign key value must be wholly null.

Relational Keys

Taking an example of Employee table:

Employee (EmployeeID, FullName, PassportNo, DeptID)

- **Candidate key.** Are individual columns in a table that qualifies for uniqueness of all the rows. In Employee table EmployeeID and PassportNo are candidate keys.

- **Primary key.** The column that is chosen to maintain uniqueness in a table. In Employee table, you can choose either EmployeeID or PassportNo columns. EmployeeID is preferable choice.
- **Alternate keys**. Candidate column other than the Primary key column, like if EmployeeID is the PK, then PassportNo would be the alternate key.
- **Super key.** If you add any other column/attribute to a primary key, then it becomes a super key, e.g EmployeeID + FullName is a super key.
- **Foreign key.** An attribute, or set of attributes, within one relation that matches the candidate key of some relation.
- **Composite key** A candidate key that consists of two or more attributes. In some cases, the key of an entity type is composed of several attributes, whose values together are unique for each entity occurrence but not separately.

RELATIONAL DATABASE LANGUAGE

Structured Query Language (SQL)

This is a language that emerged from the development of the relational model.

SQL components:

i) Data Definition Language (DDL) - Includes commands to create database objects such as tables and views, as well as commands to define access rights to those database objects i.e DDL provides commands for defining relation schemes, deletion relation

Example DDL commands include:

COMMAND DESCRIPTION

CREATE DATABASE - Creates a database schema
CREATE TABLE - Creates a new table in the user's database schema
NOT NULL - Ensures that a column will not have null values
PRIMARY KEY - Defines a primary key for a table
FOREIGN KEY - Defines a foreign key for a table
DROP TABLE - Permanently deletes a table (and its data)

ii) Data Manipulation Language (DML) for retrieving and updating data. It includes commands to insert tuples into, delete tuples from and modify tuples in the database.

Example DML commands include:

COMMAND DESCRIPTION

INSERT - Inserts row(s) into a table

SELECT - Selects attributes from rows in one or more tables or views

WHERE - Restricts the selection of rows based on a conditional expression

UPDATE - Modifies an attribute ’s values in one or more table ’s rows

DELETE - Deletes one or more rows from a table

COMPARISON OPERATORS

=, <, >, <=, >=, <> - Used in conditional expressions

AGGREGATE FUNCTIONS Used with SELECT to return mathematical summaries on columns

COUNT - Returns the number of rows with non-null values for a given column

MIN - Returns the minimum attribute value found in a given column

MAX - Returns the maximum attribute value found in a given column

SUM - Returns the sum of all values for a given column

AVG - Returns the average of all values for a given column

Writing SQL Commands

An SQL statement consists of **reserved words** and **user-defined words**.

i) Reserved words are a fixed part of the SQL language and have a fixed meaning. They must be spelt *exactly* as required and cannot be split across lines.
ii) User-defined words are made up by the user (according to certain syntax rules) and represent the names of various database objects such as tables, columns, views and so on.

- SQL require the use of a statement terminator to mark the end of each SQL statement (usually the semicolon ';' is used).
- Most components of an SQL statement are **case insensitive**. However character data must be typed *exactly* as it appears in the database. For example, if you store a person's surname as 'JANE' and then search for it using the string 'Jane', the row will not be found.

Basic Structure of SQL Statement

Basic structure of an SQL expression consists of 3 clauses;

i. SELECT

ii. FROM

iii. WHERE

SELECT

It specifies which attributes are to appear in the output.

FROM

Specifies the table or tables to be used

WHERE

It consist of a predicate involving attributes of the relations that appear in the **FROM** clause.

Given the Branch and Staff relations below.

Branch

branchNo	street	city	postcode
B001	32 Moi Avenue	Nirobi	SW1 4EH
B002	45 Clover	Kigali	AB2 3SU
B003	21 Manse	Bujumbura	G11 9QX
B004	88 Sudi Road	Kisumu	BS99 1NZ

Or

Branch (branchNo, street, city, postcode)

Staff

staffNo	fName	lName	position	sex	DOB	salary	branchNo
S001	Jane	Karimi	Manager	F	1-Oct-45	30000	B002
S002	Harry	Wanjala	Assistant	M	10-Nov-60	12000	B003
S003	Joy	Auma	Supervisor	F	24-Mar-58	18000	B001
S004	Henry	Abdala	Assistant	M	19-Feb-70	9000	B001
S005	Ian	Sawe	Manager	M	3-Jun-40	24000	B001

Or

Staff (staffNo, fName, lName, position, sex, DOB, salary, branchNo)

1. ***Retrieve all columns, all rows***

List full details of all staff.

SELECT staffNo, fName, lName, position, sex, DOB, salary, branchNo

FROM Staff;

Or

SELECT *

FROM Staff;

2. ***Retrieve specific columns, all rows***

Produce a list of salaries for all staff, showing only the staff number, the first and last names, and the salary details.

SELECT staffNo, fName, lName, salary

FROM Staff;

Row selection (WHERE clause)

3. ***Comparison search condition***

List all staff with a salary greater than 10,000.

SELECT *

FROM Staff

WHERE salary > 10000;

The selection creates a new table containing only those Staff rows with a salary greater than 10,000.

4. ***Compound comparison search condition***

List the details of all branch offices in Nairobi or Bujumbura.

SELECT *

FROM Branch

WHERE city = ‘Nairobi’ **OR** city = ‘Bujumbura’;

5. ***Range search condition***

List all staff with a salary between Kshs. 30,000 and Kshs. 40,000.

SELECT *

FROM Staff

WHERE salary > 30000 **AND** salary < 40000;

6. ***Set membership search condition***

List all managers and assistants.

SELECT *

FROM Staff

WHERE position = 'Manager' **OR** position = 'Assistant';

Using the SQL Aggregate Functions

- COUNT – returns the number of values in a specified column;
- SUM – returns the sum of the values in a specified column;
- AVG – returns the average of the values in a specified column;
- MIN – returns the smallest value in a specified column;
- MAX – returns the largest value in a specified column.

7. ***Use of COUNT and SUM***

Find the total number of Managers and the sum of their salaries.

SELECT COUNT (staffNo) **AS** myCount, **SUM**(salary) **AS** mySum

FROM Staff

WHERE position = 'Manager';

myCount	mySum
2	54000

8. ***Use of MIN, MAX, AVG***

Find the minimum, maximum, and average staff salary.

SELECT MIN (salary) **AS** myMin, **MAX**(salary) **AS** myMax, **AVG**(salary) **AS** myAvg

FROM Staff;

myMin	myMax	myAvg
9000	30000	17000

Modification of the Database

Involves **ADD**, **REMOVE** or **CHANGE** of information in the database.

(i) Deletion

DELETE FROM r

WHERE P

- P represents the column, r represent the relation.
- The statement first finds all tuples t in r which P(t) is true & then deletes them from r

Where clause can be omitted in which case all tuples in relation are deleted.

Example

To delete all tuples in Staff relation, use

DELETE FROM Staff;

- Deletes all tuples from the Staff relation.

To delete all Staff with salaries between Kshs. 10000 and Kshs. 20000, use

DELETE FROM Staff

WHERE salary > 10000 **AND** salary < 20000;

(ii) **Insertion**

To insert data into a relation:-

- Specify a tuple to be inserted or
- Write a query whose result is a set of tuples to be inserted

Tuples to be inserted must be in the correct order.

Example

Insert a new row into the Staff *table supplying data for all columns.*

INSERT INTO Staff

VALUES (‘S010’, ‘June’, ‘Moraa’, ‘Assistant’, ‘F’, ‘1970-05-25’, ‘13000’,

‘B004’);

(iii) **Updates**

To change a value in a tuple without changing all values the **UPDATE** statement can be used.

a. Give all staff a 10% pay increase.

UPDATE Staff

SET salary = salary*1.10;

UPDATE specific rows

b. Give all Managers a 20% pay increase.

UPDATE Staff

SET salary = salary*1.20

WHERE position = ‘Manager’;

UPDATE multiple columns

c. Promote Harry Wanjala (staffNo = ‘S002’) to Manager and change his salary to Kshs.100,000.

UPDATE Staff

SET position = ‘Manager’, salary = 100,000

WHERE staffNo = ‘S005’;

Granting and Revoking Privileges

i. GRANT all privileges

Give the user with authorization identifier Manager full privileges to the Staff table.

GRANT ALL PRIVILEGES

ON Staff

TO Manager;

• The user identified as Manager can retrieve rows from the Staff table, and also insert, update, and delete data from this table.

ii. GRANT specific privileges

Give users Manager and Assistant the privileges SELECT and UPDATE on column salary of the Staff table.

```
GRANT SELECT, UPDATE (salary)
ON Staff
TO Manager, Assistant;
```

iii. GRANT specific privileges to PUBLIC

Give all users the privilege SELECT on the Branch table.

```
GRANT SELECT
ON Branch
TO PUBLIC;
```

• The use of the keyword PUBLIC means that all users are able to retrieve all the data in the Branch table.

Revoking Privileges from Users (REVOKE)

• The REVOKE statement is used to take away privileges that were granted with the GRANT statement.

• A REVOKE statement can take away all or some of the privileges that were previously granted to a user. The format of the statement is:

iv. REVOKE specific privileges from PUBLIC

Revoke the privilege SELECT on the Branch table from all users.

```
REVOKE SELECT
ON Branch
FROM PUBLIC;
```

v. REVOKE specific privileges from named user

Revoke all privileges you have given to Assistant on the Staff table.

REVOKE ALL PRIVILEGES

ON Staff

FROM Assistant;

DATA DEFINITION LANGUAGE

1. **Table Definition in SQL**

Syntax

CREATE TABLE TableName

(columnName1 dataType [NOT NULL], columnName2 dataType, …, columnNamen dataType);

Example1

CREATE TABLE Staff

(staffNo **CHAR** (5) **NOT NULL**,

fName **CHAR** (10),

lName **CHAR** (10),

position **CHAR** (20),

sex **CHAR** (5),

DOB **DATE**,

salary **INTEGER**(6),

branchNo CHAR (5),

PRIMARY KEY(staffNo),

FOREIGN KEY(branchNo));

Example2

CREATE TABLE Branch

(branchNo CHAR (5) NOT NULL,

street CHAR (20) NOT NULL,

city CHAR (20) NOT NULL,

postcode CHAR (20) NOT NULL)

PRIMARY KEY (branchNo));

2. **Creating Database in SQL**

Syntax

CREATE DATABASE *dbname*;

Before entering tables in a database, you have to declare it usable. The syntax used is:

USE *dbname*;