Final Exam Cheatsheet

Monday, August 5, 2024 8:25 PM

Chapter 1: Databases and Database Users

While this chapter is more theoretical, here are some basic SQL statements to get started:

```
-- Create a simple table
CREATE TABLE Employee (
    EmployeeID INT PRIMARY KEY,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    BirthDate DATE,
    HireDate DATE
);

-- Insert a record into the Employee table
INSERT INTO Employee (EmployeeID, FirstName, LastName, BirthDate, HireDate)
VALUES (1, 'John', 'Doe', '1980-01-01', '2020-01-01');

-- Select all records from the Employee table
SELECT * FROM Employee;
```

Chapter 2: Database System Concepts and Architecture

Basic schema creation and manipulation:

```
-- Create a table with different data types
CREATE TABLE Department (
    DepartmentID INT PRIMARY KEY,
    DepartmentName VARCHAR(100) NOT NULL
);

-- Alter the table to add a new column
ALTER TABLE Department ADD Location VARCHAR(50);

-- Drop a column from the table
ALTER TABLE Department DROP COLUMN Location;

-- Drop the table
DROP TABLE Department;
```

Chapter 3: Data Modeling Using the Entity-Relationship (ER) Model

Examples of creating tables with relationships:

```
-- Create Department and Employee tables with a foreign key relationship
CREATE TABLE Department (
    DepartmentID INT PRIMARY KEY,
    DepartmentName VARCHAR(100) NOT NULL
);

CREATE TABLE Employee (
    EmployeeID INT PRIMARY KEY,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    BirthDate DATE,
    HireDate DATE,
    DepartmentID INT,
    FOREIGN KEY (DepartmentID) REFERENCES Department(DepartmentID)
);

-- Insert records into the tables
INSERT INTO Department (DepartmentID, DepartmentName)
VALUES (1, 'Human Resources'), (2, 'Engineering');

INSERT INTO Employee (EmployeeID, FirstName, LastName, BirthDate, HireDate, DepartmentID)
VALUES (1, 'John', 'Doe', '1980-01-01', '2020-01-01', 1),
       (2, 'Jane', 'Smith', '1985-05-15', '2021-06-01', 2);

-- Select records with join
SELECT e.FirstName, e.LastName, d.DepartmentName
FROM Employee e
JOIN Department d ON e.DepartmentID = d.DepartmentID;
```

Chapter 4: SQL-99: Schema Definition, Constraints, Queries, and Views

Defining schemas, constraints, and basic queries:

```
-- Create table with constraints
CREATE TABLE Project (
    ProjectID INT PRIMARY KEY,
    ProjectName VARCHAR(100) NOT NULL,
    StartDate DATE,
    EndDate DATE CHECK (EndDate > StartDate)
);

-- Insert records into the Project table
INSERT INTO Project (ProjectID, ProjectName, StartDate, EndDate)
VALUES (1, 'Project Alpha', '2022-01-01', '2022-12-31');

-- Update a record
UPDATE Project SET EndDate = '2023-12-31' WHERE ProjectID = 1;

-- Delete a record
DELETE FROM Project WHERE ProjectID = 1;

-- Select with basic query
SELECT * FROM Project;
```

Chapter 5: Complex SQL

Examples of complex SQL queries:

```
-- Correlated subquery: Find employees who earn more than the average salary in their department
SELECT e.FirstName, e.LastName
FROM Employee e
WHERE e.Salary > (
    SELECT AVG(e2.Salary)
    FROM Employee e2
    WHERE e2.DepartmentID = e.DepartmentID
);

-- Using IN and EXISTS
SELECT FirstName, LastName
FROM Employee
WHERE DepartmentID IN (SELECT DepartmentID FROM Department WHERE DepartmentName = 'Engineering');

-- Aggregate functions and GROUP BY
SELECT DepartmentID, AVG(Salary) AS AvgSalary
FROM Employee
GROUP BY DepartmentID
HAVING AVG(Salary) > 50000;

-- Subquery in FROM clause
SELECT d.DepartmentName, emp_count
FROM Department d,
    (SELECT DepartmentID, COUNT(*) AS emp_count
     FROM Employee
     GROUP BY DepartmentID) e
WHERE d.DepartmentID = e.DepartmentID;

-- Grant and revoke permissions
GRANT SELECT, INSERT ON Employee TO user_name;
REVOKE INSERT ON Employee FROM user_name;
```

```
-- Grant and revoke permissions
GRANT SELECT, INSERT ON Employee TO user_name;
REVOKE INSERT ON Employee FROM user_name;
```