CS 411 STAGE 3: Database Design

AI-Powered Job Recommendation and Notification System (Job-Genie)

Devansh Agarwal, Aaryan Gusain, Nakul Kuppu

Below is the Data Definition Language (DDL) commands:

```
CREATE TABLE User (
  UserID INT PRIMARY KEY AUTO_INCREMENT,
  Email VARCHAR(255) UNIQUE NOT NULL,
  Name VARCHAR(255) NOT NULL,
  Resume TEXT,
  Preferences TEXT
);
CREATE TABLE Job (
  JobID INT PRIMARY KEY AUTO_INCREMENT,
  Title VARCHAR(255) NOT NULL,
  Company VARCHAR(255),
  Location VARCHAR(255),
  Salary DECIMAL(10,2),
  Description TEXT,
  Source VARCHAR(255)
);
CREATE TABLE JobMatch (
  MatchID INT PRIMARY KEY AUTO_INCREMENT,
  UserID INT NOT NULL,
  JobID INT NOT NULL,
```

```
  MatchScore DECIMAL(5,2) NOT NULL,
  Timestamp DATETIME DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (UserID) REFERENCES User(UserID) ON DELETE CASCADE,
  FOREIGN KEY (JobID) REFERENCES Job(JobID) ON DELETE CASCADE
);
CREATE TABLE Application (
  ApplicationID INT PRIMARY KEY AUTO_INCREMENT,
  UserID INT NOT NULL,
  JobID INT NOT NULL,
  NotificationID INT NOT NULL,
  Status VARCHAR(50) NOT NULL CHECK (Status IN ('Pending', 'Accepted', 'Rejected')),
  AppliedAt DATETIME DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (UserID) REFERENCES User(UserID) ON DELETE CASCADE,
  FOREIGN KEY (JobID) REFERENCES Job(JobID) ON DELETE CASCADE,
  FOREIGN KEY (NotificationID) REFERENCES Notification(NotificationID) ON DELETE
CASCADE
);
CREATE TABLE Notification (
  NotificationID INT PRIMARY KEY AUTO_INCREMENT,
  UserID INT NOT NULL,
  JobID INT NOT NULL,
  ApplicationID INT NOT NULL,
  Status VARCHAR(255) NOT NULL,
  SentAt DATETIME DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (UserID) REFERENCES User(UserID) ON DELETE CASCADE,
```

```
  FOREIGN KEY (JobID) REFERENCES Job(JobID) ON DELETE CASCADE,
  FOREIGN KEY (ApplicationID) REFERENCES Application(ApplicationID) ON DELETE
CASCADE
);
CREATE TABLE UserJob (
  UserID INT,
  JobID INT,
  PRIMARY KEY (UserID, JobID),
  FOREIGN KEY (UserID) REFERENCES User(UserID) ON DELETE CASCADE,
  FOREIGN KEY (JobID) REFERENCES Job(JobID) ON DELETE CASCADE
);
```

Advanced Queries and and Screenshots:

Query 1: Recent Applications (Last 7 Days)

```
SELECT
    a.ApplicationID,
    u.Name,
    j.Title,
    a.AppliedAt
FROM
    Application a
JOIN
    User u ON a.UserID = u.UserID
JOIN
    Job j ON a.JobID = j.JobID
WHERE
    a.AppliedAt >= NOW() - INTERVAL 7 DAY
ORDER BY
    a.AppliedAt DESC;
```

RESULTS

ApplicationID	Name	Title	AppliedAt
997	Unknown	Unknown	2025-04-02 03:54:35
998	Unknown	Unknown	2025-04-02 03:54:35
999	Unknown	Unknown	2025-04-02 03:54:35
1000	Unknown	Unknown	2025-04-02 03:54:35
986	Unknown	Unknown	2025-04-02 03:54:34
987	Unknown	Unknown	2025-04-02 03:54:34
988	Unknown	Unknown	2025-04-02 03:54:34
989	Unknown	Unknown	2025-04-02 03:54:34
990	Unknown	Unknown	2025-04-02 03:54:34
991	Unknown	Unknown	2025-04-02 03:54:34
992	Unknown	Unknown	2025-04-02 03:54:34
993	Unknown	Unknown	2025-04-02 03:54:34
994	Unknown	Unknown	2025-04-02 03:54:34
995	Unknown	Unknown	2025-04-02 03:54:34
996	Unknown	Unknown	2025-04-02 03:54:34
972	Unknown	Unknown	2025-04-02 03:54:33
973	Unknown	Unknown	2025-04-02 03:54:33
974	Unknown	Unknown	2025-04-02 03:54:33
975	Unknown	Unknown	2025-04-02 03:54:33
976	Unknown	Unknown	2025-04-02 03:54:33

Query 2: Most Recent Application per User

```
SELECT
    u.UserID,
    u.Name,
    j.JobID,
    j.Title,
    a.AppliedAt
FROM
    User u
JOIN
    Application a ON u.UserID = a.UserID
JOIN
    Job j ON a.JobID = j.JobID
WHERE
    a.AppliedAt = (
        SELECT MAX(a2.AppliedAt)
        FROM Application a2
        WHERE a2.UserID = u.UserID
    );
```

RESULTS

UserID	Name	JobID	Title	AppliedAt
884	Unknown	1	Unknown	2025-04-02 03:53:21
885	Unknown	382	Unknown	2025-04-02 03:53:49
886	Unknown	532	Unknown	2025-04-02 03:54:00
887	Unknown	979	Unknown	2025-04-02 03:54:33
888	Unknown	264	Unknown	2025-04-02 03:53:40
889	Unknown	709	Unknown	2025-04-02 03:54:12
890	Unknown	197	Unknown	2025-04-02 03:53:36
891	Unknown	580	Unknown	2025-04-02 03:54:03
892	Unknown	9	Unknown	2025-04-02 03:53:21
893	Unknown	901	Unknown	2025-04-02 03:54:26
894	Unknown	11	Unknown	2025-04-02 03:53:22
895	Unknown	12	Unknown	2025-04-02 03:53:22
896	Unknown	864	Unknown	2025-04-02 03:54:24
897	Unknown	854	Unknown	2025-04-02 03:54:23
898	Unknown	15	Unknown	2025-04-02 03:53:22
899	Unknown	422	Unknown	2025-04-02 03:53:52
900	Unknown	967	Unknown	2025-04-02 03:54:32
901	Unknown	18	Unknown	2025-04-02 03:53:22
902	Unknown	19	Unknown	2025-04-02 03:53:22
903	Unknown	470	Unknown	2025-04-02 03:53:56

Query 3: Users Who Applied to More Than One Job per Company

```
SELECT
    u.UserID,
    u.Name,
    j.Company,
    COUNT(DISTINCT j.JobID) AS NumApplications
FROM
    User u
JOIN
    Application a ON u.UserID = a.UserID
JOIN
    Job j ON a.JobID = j.JobID
GROUP BY
    u.UserID, j.Company, u.Name
HAVING
    COUNT(DISTINCT j.JobID) > 1;
```

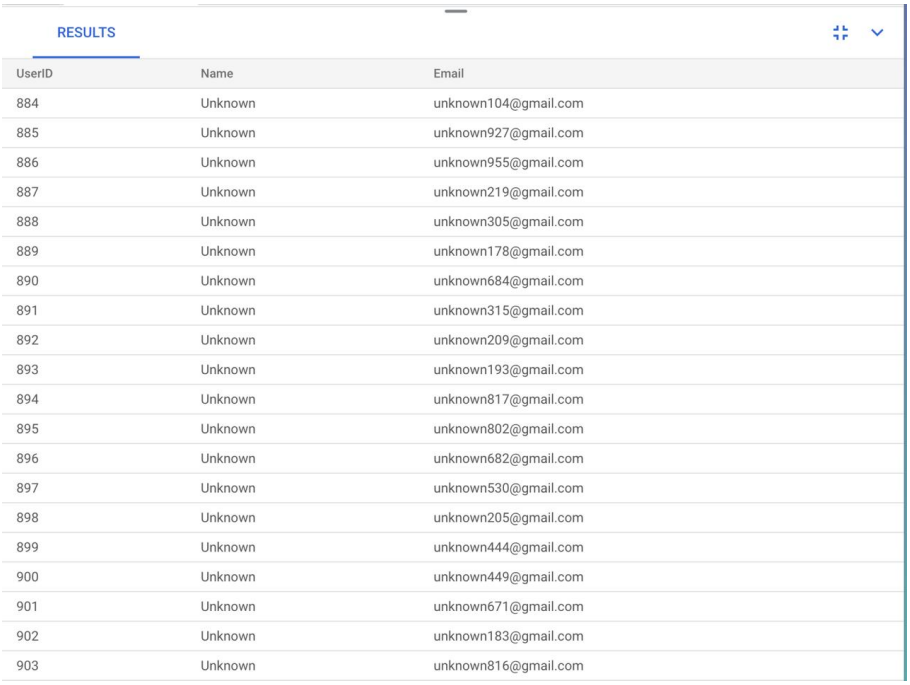
RESULTS

UserID	Name	Company	NumApplications
885	Unknown	Unknown	3
886	Unknown	Unknown	2
887	Unknown	Unknown	2
888	Unknown	Unknown	3
889	Unknown	Unknown	5
890	Unknown	Unknown	2
891	Unknown	Unknown	2
893	Unknown	Unknown	2
896	Unknown	Unknown	4
897	Unknown	Unknown	4
899	Unknown	Unknown	2
900	Unknown	Unknown	3
903	Unknown	Unknown	3
904	Unknown	Unknown	3
906	Unknown	Unknown	4
907	Unknown	Unknown	2
909	Unknown	Unknown	6
913	Unknown	Unknown	2
914	Unknown	Unknown	2
915	Unknown	Unknown	2

Query 4: Users Who Didn’t Apply to the Highest Salary Job

```
SELECT
    DISTINCT u.UserID,
    u.Name,
    u.Email
FROM
    User u
JOIN

    Application a ON u.UserID = a.UserID
WHERE
    u.UserID NOT IN (
        SELECT a.UserID
        FROM Application a
        JOIN Job j ON a.JobID = j.JobID
        WHERE j.Salary = (SELECT MAX(Salary) FROM Job)
    );
```



RESULTS

UserID	Name	Email
884	Unknown	unknown104@gmail.com
885	Unknown	unknown927@gmail.com
886	Unknown	unknown955@gmail.com
887	Unknown	unknown219@gmail.com
888	Unknown	unknown305@gmail.com
889	Unknown	unknown178@gmail.com
890	Unknown	unknown684@gmail.com
891	Unknown	unknown315@gmail.com
892	Unknown	unknown209@gmail.com
893	Unknown	unknown193@gmail.com
894	Unknown	unknown817@gmail.com
895	Unknown	unknown802@gmail.com
896	Unknown	unknown682@gmail.com
897	Unknown	unknown530@gmail.com
898	Unknown	unknown205@gmail.com
899	Unknown	unknown444@gmail.com
900	Unknown	unknown449@gmail.com
901	Unknown	unknown671@gmail.com
902	Unknown	unknown183@gmail.com
903	Unknown	unknown816@gmail.com

Query 5: Users with More Applications than the Average

```
SELECT
    u.UserID,
    COUNT(a.ApplicationID) AS TotalApplications
FROM User u
JOIN Application a ON u.UserID = a.UserID
GROUP BY u.UserID
HAVING COUNT(a.ApplicationID) >= (
    SELECT AVG(app_count)
    FROM (
        SELECT COUNT(*) AS app_count
        FROM Application
        GROUP BY UserID
    ) AS sub
);
```

RESULTS

UserID	TotalApplications
885	3
886	2
887	2
888	3
889	5
890	2
891	2
893	2
896	4
897	4
899	2
900	3
903	3
904	3
906	4
907	2
909	6
913	2
914	2
915	2

Count Query to show 1000 in 3 tables:

```
SELECT
    (SELECT COUNT(*) FROM Job) AS JobCount,
    (SELECT COUNT(*) FROM Application) AS ApplicationCount,
    (SELECT COUNT(*) FROM User) AS UserCount;
```