SQL FOUNDATION

INTRODUCTION TO RELATIONAL DATABASE

OBJECTIVES

- SQL with Relational Databases
- Retrieve data
- Database tables
- Filter data
- Group data

SQL OVERVIEW

- Structured Query Language
- SQL standard
  - Database extensions

SQL LANGUAGE GROUPING

- Data Manipulation Language (DML)
- Data Definition Language (DDL)
- Data Control Language (DCL)

RELATIONAL BASICS

- Employees

Employee_ID	First_Name	Last_Name	Dept_ID	Location_ID
1001	John	Jones	10	100
1002	Susan	Smith	20	100
1003	Jackson	Black	10	200
1004	Thom	Thomas	20	300
1005	Robert	Reid	10	400

RELATIONSHIPS

- Employees and Departments

Employee_ID	First_Name	Last_Name	Dept_ID	Location_ID
1001	John	Jones	10	100
1002	Susan	Smith	20	100
1003	Jackson	Black	10	200
1004	Thom	Thomas	20	300
1005	Robert	Reid	10	400

Dept_ID	Name
10	Human Resources
20	Sales

RELATIONSHIPS

- Employees and JobHistories

Employee_ID	First_Name	Last_Name	Dept_ID	Location_ID
1001	John	Jones	10	100
1002	Susan	Smith	20	100
1003	Jackson	Black	10	200
1004	Thom	Thomas	20	300
1005	Robert	Reid	10	400

Employee_ID	Position_ID	Start_Date	End_Date
1005	2011	20180824	20200105
1005	2015	20200106	NULL

SELECT STATEMENT

SELECT column1 [, column2 …] FROM tablename

For example

SELECT * FROM Employees;

SELECT First_Name, Last_Name FROM Employees;

DISTINCT SELECT STATEMENT

SELECT DISTINCT column1 FROM tablename

For example

SELECT DISTINCT Last_Name FROM Employees;

LITERAL STRINGS

- Single quotes are most commonly used

JOINING TABLES

```
SELECT First_Name, Last_Name, Name
FROM Employees JOIN Departments
ON Employees.Dept_ID = Departments.Dept_ID;
```

```
SELECT First_Name, Last_Name, Name
FROM Employees LEFT JOIN Departments
ON Employees.Dept_ID = Departments.Dept_ID;
```

FILTERING

```
SELECT First_Name, Last_Name, Name
FROM Employees JOIN Departments
ON Employees.Dept_ID = Departments.Dept_ID
WHERE Last_Name = ‘Smith’;
```

SORTING DATA

```
SELECT First_Name, Last_Name, Name
FROM Employees JOIN Departments
ON Employees.Dept_ID = Departments.Dept_ID
WHERE Last_Name = ‘Smith’
ORDER BY Name DESC;
```

AGGREGATING DATA

```
SELECT COUNT(Location_ID) FROM Employees;
```

GROUPING DATA

```
SELECT COUNT(Employee_ID), Name FROM Employees
JOIN Departments
ON Employees.Dept_ID = Departments.Dept_ID;

SELECT COUNT(Employee_ID), Name FROM Employees
JOIN Departments
ON Employees.Dept_ID = Departments.Dept_ID
GROUP BY Name;
```

SELECTING GROUPS

```
SELECT COUNT(Employee_ID), Name FROM Employees
JOIN Departments
ON Employees.Dept_ID = Departments.Dept_ID
GROUP BY Name
HAVING COUNT(Employee_ID) > 10;
```

THANK YOU