Week 12

Database Systems -

Introduction to Databases and Data Warehouses

# CHAPTER 4 - Update Operations, Update Anomalies, and Normalization (Part 3 & 4)

# MAIN TOPICS

- Normalization Example 2 – University Recruiting (cont'd)
- Case Study – Exercise 4.5
  - Update Anomaly, Functional Dependency, Normalization

# Normalization
# Example 2: relation RECRUITING

Table columns and data

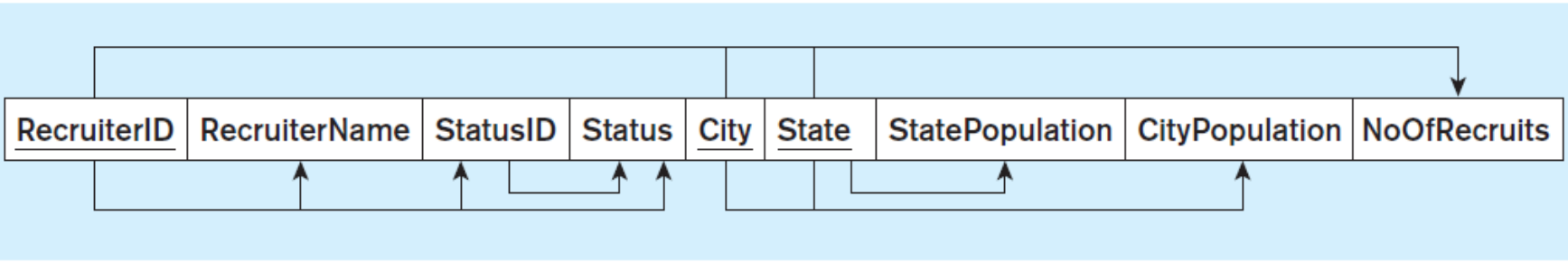
RECRUITING

| RecruiterID | RecruiterName | StatusID | Status | City | State | StatePopulation | CityPopulation | NoOfRecruits |
|---|---|---|---|---|---|---|---|---|
| R1 | Katy | IF | Internal Full Time | Portland | ME | 1,350,000 | 70,000 | 11 |
| R1 | Katy | IF | Internal Full Time | Grand Rapids | MI | 9,900,000 | 190,000 | 20 |
| R2 | Abra | IP | Internal Part Time | Rockford | IL | 12,900,000 | 340,000 | 17 |
| R3 | Jana | CN | Contractor | Spokane | WA | 6,800,000 | 210,000 | 8 |
| R3 | Jana | CN | Contractor | Portland | OR | 3,900,000 | 600,000 | 30 |
| R3 | Jana | CN | Contractor | Eugene | OR | 3,900,000 | 360,000 | 20 |
| R4 | Maria | IF | Internal Full Time | Rockford | IL | 12,900,000 | 340,000 | 14 |
| R4 | Maria | IF | Internal Full Time | Grand Rapids | MN | 5,400,000 | 11,000 | 9 |
| R5 | Dan | CN | Contractor | Grand Rapids | MI | 9,900,000 | 190,000 | 33 |

Composite primary key: (RecruiterID, City, State)
Each row: a specific recruiter's recruiting result in one city.

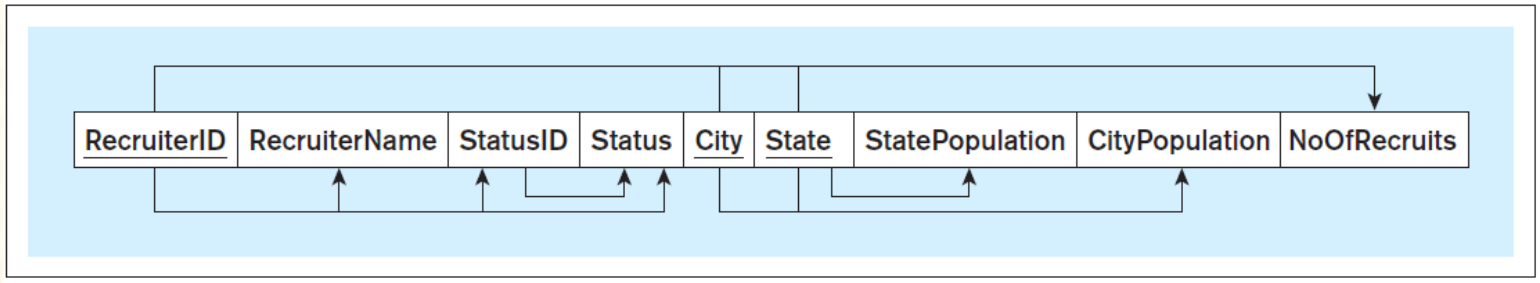
# Normalization Example 2: Normalizing a table to 2NF

1. identify the types of functional dependencies.



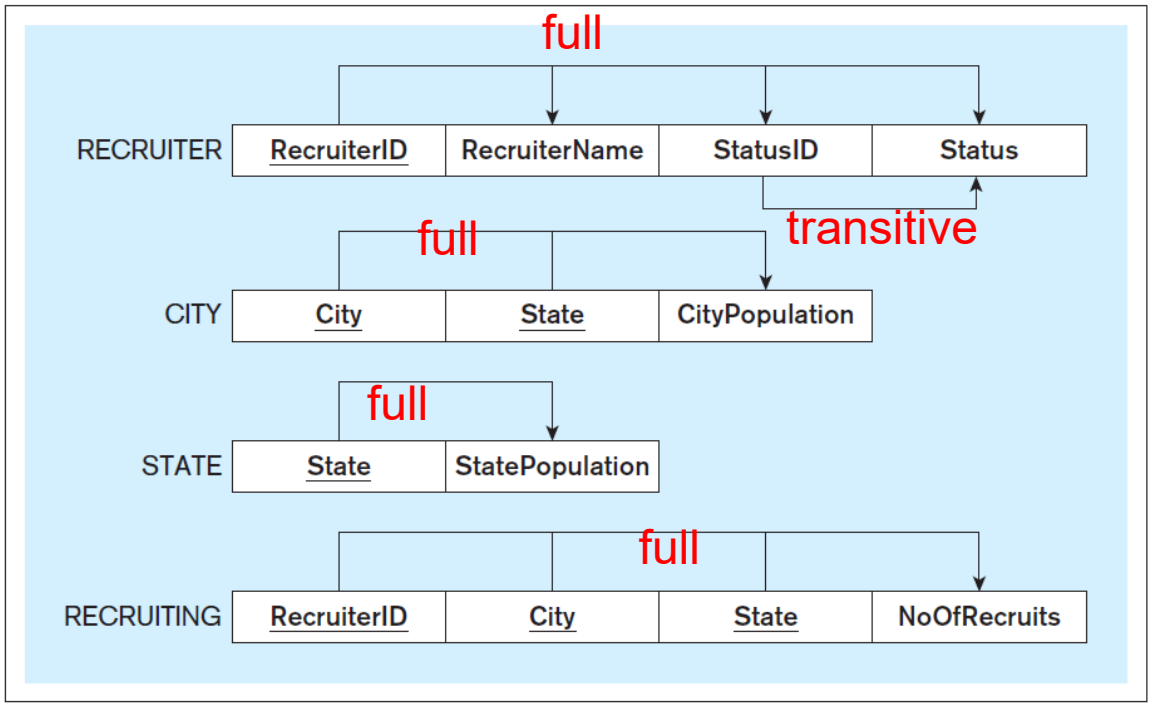

| Type | Functional Dependency |
|---|---|
| Partial | RecruiterID -> RecuriterName, StatusID, Status |
| Transitive | StatusID -> Status |
| Partial | City, State -> CityPopulation |
| Partial | State -> StatePopulation |
| Full | RecruiterID, City, State -> NoOfRecruits |

# Normalization Example 2 : Normalizing a table to 2NF




1NF

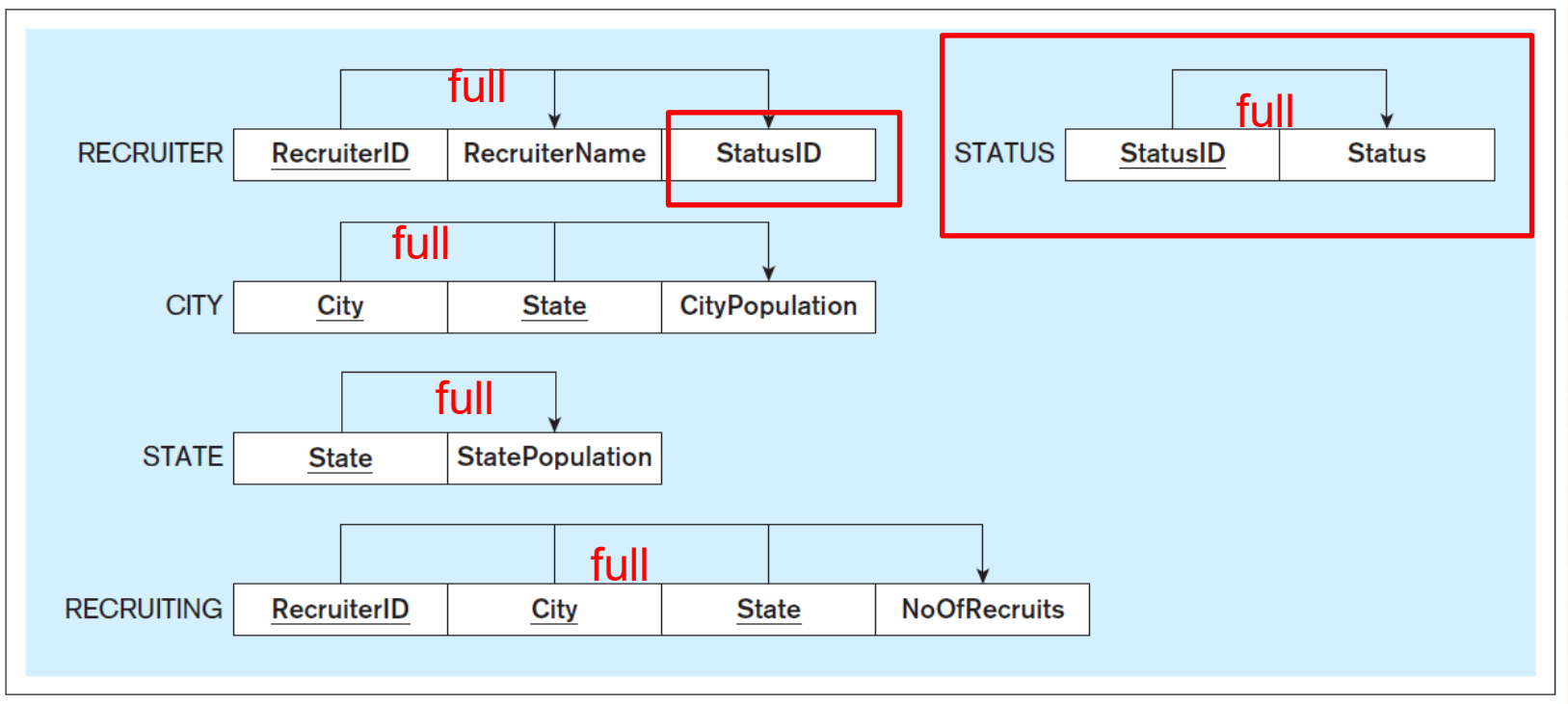
Nnormalized to 2NF




2. Remove partial functional dependencies (FDs) by adding 1 extra able for each set of partial FDs.

# Normalization Example 2 : Normalizing a table to 3NF

3NF




Remove transitive functional dependencies (FDs) by adding 1 extra able for each set of transitive FDs.

# Central Plane University relation RECRUITING – not normalized, prone to update anomalies

RECRUITING

| RecruiterID | RecruiterName | StatusID | Status | City | State | StatePopulation | CityPopulation | NoOfRecruits |
|---|---|---|---|---|---|---|---|---|
| R1 | Katy | IF | Internal Full Time | Portland | ME | 1,350,000 | 70,000 | 11 |
| R1 | Katy | IF | Internal Full Time | Grand Rapids | MI | 9,900,000 | 190,000 | 20 |
| R2 | Abra | IP | Internal Part Time | Rockford | IL | 12,900,000 | 340,000 | 17 |
| R3 | Jana | CN | Contractor | Spokane | WA | 6,800,000 | 210,000 | 8 |
| R3 | Jana | CN | Contractor | Portland | OR | 3,900,000 | 600,000 | 30 |
| R3 | Jana | CN | Contractor | Eugene | OR | 3,900,000 | 360,000 | 20 |
| R4 | Maria | IF | Internal Full Time | Rockford | IL | 12,900,000 | 340,000 | 14 |
| R4 | Maria | IF | Internal Full Time | Grand Rapids | MN | 5,400,000 | 11,000 | 9 |
| R5 | Dan | CN | Contractor | Grand Rapids | MI | 9,900,000 | 190,000 | 33 |

# Central Plane University example - normalized relations with data (redundancy eliminated and update anomalies resolved)

RECRUITER

| RecruiterID | RecruiterName | StatusID |
|---|---|---|
| R1 | Katy | IF |
| R2 | Abra | IP |
| R3 | Jana | CN |
| R4 | Maria | IF |
| R5 | Dan | CN |

STATUS

| StatusID | Status |
|---|---|
| CN | Contractor |
| IF | Internal Full Time |
| IP | Internal Part Time |

CITY

| City | State | CityPopulation |
|---|---|---|
| Portland | ME | 70,000 |
| Grand Rapids | MI | 190,000 |
| Rockford | IL | 340,000 |
| Spokane | WA | 210,000 |
| Portland | OR | 600,000 |
| Eugene | OR | 360,000 |
| Grand Rapids | MN | 11,000 |

STATE

| State | StatePopulation |
|---|---|
| ME | 1,350,000 |
| MI | 9,900,000 |
| IL | 12,900,000 |
| WA | 6,800,000 |
| OR | 3,900,000 |
| MN | 5,400,000 |

RECRUITING

| RecruiterID | City | State | NoOfRecruits |
|---|---|---|---|
| R1 | Portland | ME | 11 |
| R1 | Grand Rapids | MI | 20 |
| R2 | Rockford | IL | 17 |
| R3 | Spokane | WA | 8 |
| R3 | Portland | OR | 30 |
| R3 | Eugene | OR | 20 |
| R4 | Rockford | IL | 14 |
| R4 | Grand Rapids | MN | 9 |
| R5 | Grand Rapids | MI | 33 |

# NORMALIZATION: Practice 2, —CHAPTER 4 END-CHAPTER EXERCISE 4.5

Lectured in class.

**E4.5** Consider the following relation with sample data.

AIRPORT KLX TABLE

| Date | AirlineID | AirlineName | TerminalID | NumberOfGates | NumberOfDepartingFlights |
|---|---|---|---|---|---|
| 11-Dec | UA | United | A | 20 | 34 |
| 11-Dec | NW | Northwest | A | 20 | 17 |
| 11-Dec | AA | American | A | 20 | 11 |
| 11-Dec | DL | Delta | B | 15 | 20 |
| 11-Dec | JB | Jet Blue | B | 15 | 6 |
| 12-Dec | UA | United | A | 20 | 29 |
| 12-Dec | DL | Delta | B | 15 | 20 |
| 12-Dec | SWA | Southwest | C | 15 | 17 |

- *The AIRPORT KLX Table captures the data about daily departing flights at the KLX Airport.*
- *Each airline operating at KLX airport has a unique Airline ID and an Airline Name.*
- *Each terminal at KLX airport has a unique Terminal ID and a fixed Number of Gates.*
- *Each airline is permanently assigned to one (and only one) terminal at the KLX Airport.*
- *Each terminal at KLX Airport can have multiple airlines assigned to it.*
- *Each day (Date), this table records the Number of Departing Flights at KLX Airport for each airline.*

**E4.5a** Using the AIRPORT KLX Table, describe an example that illustrates the insertion anomaly.

**E4.5b** Using the AIRPORT KLX Table, describe an example that illustrates the deletion anomaly.

**E4.5c** Using the AIRPORT KLX Table, describe an example that illustrates the modification anomaly.

**E4.5d** Depict full key functional dependencies, partial functional dependencies (if any), and transitive functional dependencies (if any) in the AIRPORT KLX Table.

**E4.5e** Show the result of normalizing the AIRPORT KLX Table to 2NF.

**E4.5f** Show the result of normalizing the AIRPORT KLX Table to 3NF.

**E4.5g** Using the set of tables resulting from **E4.5f**, describe how the anomalies shown in **E4.5a**, **E4.5b**, and **E4.5c** are eliminated.

# NORMALIZATION: Example 4, Practice 3 —CHAPTER 4 END-CHAPTER EXERCISE 4.6

**Lectured in class.**

**E4.6** Consider the following relation with sample data.

DEPARTMENT OF TRANSPORTATION (DOT) PROJECT TABLE

| ProjectID | Project Name | CountyID | CountyName | Project ManagerID | Project ManagerName | ProjectMiles WithinCounty |
|---|---|---|---|---|---|---|
| 1 | Road X | 1 | Wilson | M1 | Bob | 10.00 |
| 1 | Road X | 2 | Ottawa | M1 | Bob | 17.00 |
| 1 | Road X | 3 | Davis | M1 | Bob | 12.00 |
| 2 | Road Y | 3 | Davis | M2 | Sue | 23.00 |
| 3 | Bridge A | 1 | Wilson | M3 | Lee | 0.50 |
| 3 | Bridge A | 2 | Ottawa | M3 | Lee | 0.30 |
| 4 | Tunnel Q | 2 | Ottawa | M1 | Bob | 2.00 |
| 5 | Road W | 4 | Pony | M4 | Bob | 23.00 |

- *The DEPARTMENT OF TRANSPORTATION (DOT) Project Table captures the data about projects and their length (in miles).*
- *Each project has a unique Project ID and Project Name.*
- *Each county has a unique County ID and County Name.*
- *Each project manager has a unique Project Manager ID and Project Manager Name.*
- *Each project has one project manager.*
- *A project manager can manage several projects.*
- *A project can span across several counties.*
- *This table records the length of the project in a county in the ProjectMilesWithinCounty column.*

**E4.6a** Using the DOT PROJECT Table, describe an example that illustrates the insertion anomaly.

**E4.6b** Using the DOT PROJECT Table, describe an example that illustrates the deletion anomaly.

**E4.6c** Using the DOT PROJECT Table, describe an example that illustrates the modification anomaly.

**E4.6d** Depict full key functional dependencies, partial functional dependencies (if any), and transitive functional dependencies (if any) in the DOT PROJECT Table.

**E4.6e** Show the result of normalizing the DOT PROJECT Table to 2NF.

**E4.6f** Show the result of normalizing the DOT PROJECT Table to 3NF.

**E4.6g** Using the set of tables resulting from **E4.6f**, describe how the anomalies shown in **E4.6a**, **E4.6b**, and **E4.6c** are eliminated.