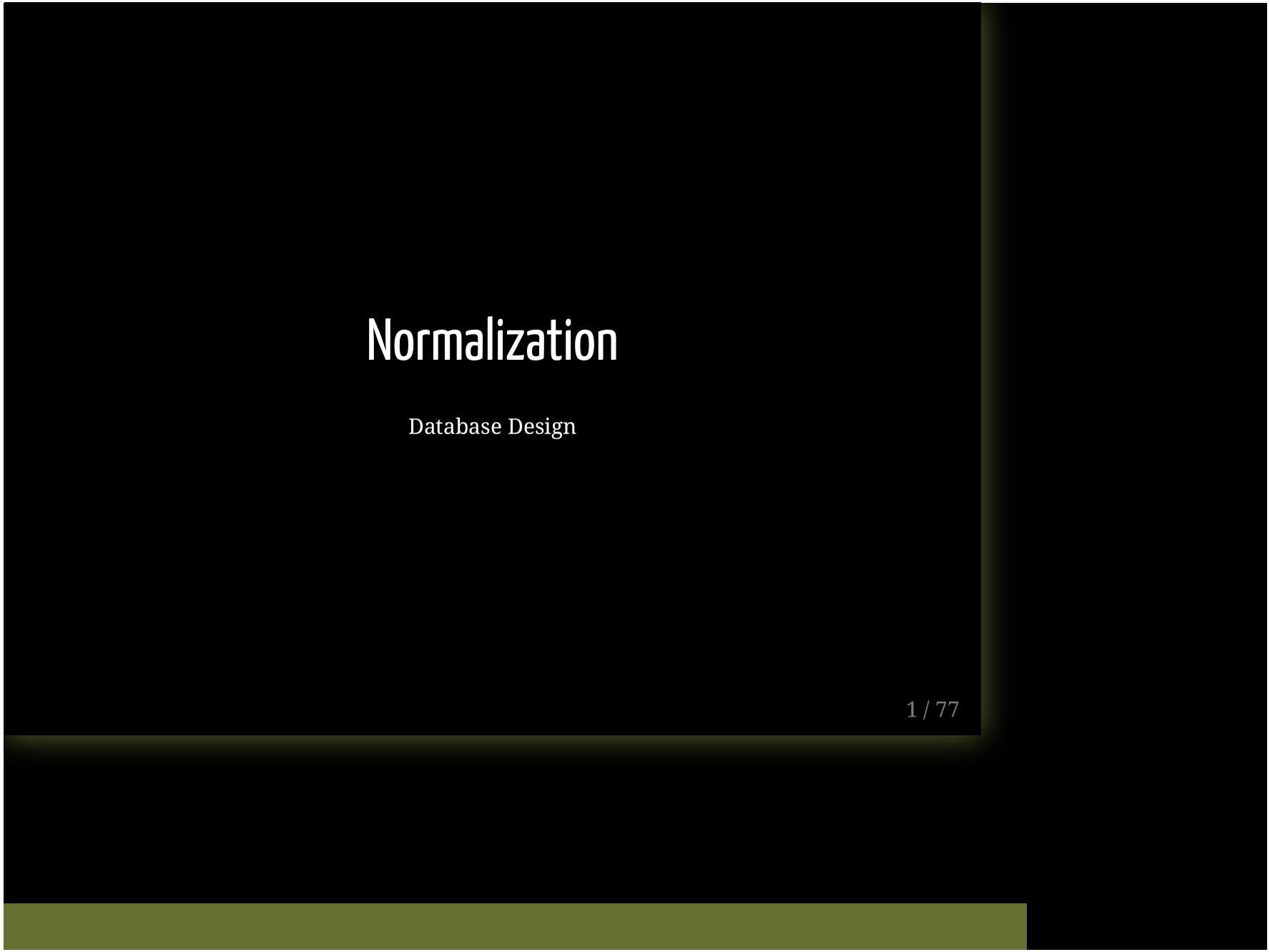
Normalization
Database Design
1 / 77

# Agenda



2 / 77

# Overview

3 / 77

# Overview

## Introduction

What are the normal forms in relational database design theory?

- Guidelines for *how to design records*.

# Overview

## Introduction

What are the normal forms in relational database design theory?

- Guidelines for *how to design records*.
- They are *general in nature*, and apply to any relational database system.

# Overview

## Introduction

What are the normal forms in relational database design theory?

- Guidelines for *how to design records*.
- They are *general in nature*, and apply to any relational database system.
- They are *designed to prevent rendundancy, ambiguity, anomalies, and data inconsistencies*.

# Overview

## Introduction

What are the normal forms in relational database design theory?

- Guidelines for *how to design records*.
- They are *general in nature*, and apply to any relational database system.
- They are *designed to prevent rendundancy, ambiguity, anomalies, and data inconsistencies*.
- They *tend to penalize retrieval* (i.e. reading with `SELECT` statements), since data which may have been retrievable from one record in an unnormalized design may have to be retrieved from several records using a **join** in the normalized form.

7 / 77

# Overview

## Introduction

What are the normal forms in relational database design theory?

- Guidelines for *how to design records*.
- They are *general in nature*, and apply to any relational database system.
- They are *designed to prevent rendundancy, ambiguity, anomalies, and data inconsistencies*.
- They *tend to penalize retrieval* (i.e. reading with `SELECT` statements), since data which may have been retrievable from one record in an unnormalized design may have to be retrieved from several records using a **join** in the normalized form.
- There is *no obligation to fully normalize* all records when actual performance requirements are taken into account.

8 / 77

# Overview

## Redundancy

One of the goals of normalization is to avoid data redundancy - the repetition of facts in multiple places within the data.

9 / 77

# Overview

## Redundancy

One of the goals of normalization is to avoid data redundancy - the repetition of facts in multiple places within the data.

An example table to hold `students` ' grades', with lots of redundancy.

| id | name | email | assessment_title | grade |
|---|---|---|---|---|
| 1 | John I. Rivero | JohnIRivero@jourrapide.com | Quiz #1 | 95 |
| 2 | John I. Rivero | JohnIRivero@jourrapide.com | Quiz #2 | 78 |
| 3 | John I. Rivero | JohnIRivero@jourrapide.com | Midterm Exam | 82 |
| ... | ... | ... | ... | ... |

10 / 77

# Overview

## Anomalies

Another goal is to avoid **anomalies**, which come in three types.

- Insertion anomalies
- Update anomalies
- Deletion anomalies

# Overview

## Insertion anomalies

Insertion anomalies occur when we are not able to insert certain attributes in the database without the presence of other attributes.

# Overview

## Insertion anomalies

Insertion anomalies occur when we are not able to insert certain attributes in the database without the presence of other attributes.

- For example, if we wanted to add a new student to our records, but couldn't do so because they hadn't yet taken any assessments, and we had mistakenly made the `assessment_title` and `grade` fields `NOT NULL`.

| id | name | email | assessment_title | grade |
|---|---|---|---|---|
| 1 | John I. Rivero | JohnIRivero@jourrapide.com | Quiz #1 | 95 |
| 2 | John I. Rivero | JohnIRivero@jourrapide.com | Quiz #2 | 78 |
| 3 | John I. Rivero | JohnIRivero@jourrapide.com | Midterm Exam | 82 |
| ... | ... | ... | ... | ... |

13 / 77

# Overview

## Update anomalies

Update anomalies occur when a correct update of a record requires other actions, such as addition, deletion or both, in order to retain data integrity.

# Overview

## Update anomalies

Update anomalies occur when a correct update of a record requires other actions, such as addition, deletion or both, in order to retain data integrity.

- For example, if we wanted to change `John I. Rivero`'s email address, but it requires us to update multiple records.

| id | name | email | assessment_title | grade |
|---|---|---|---|---|
| 1 | John I. Rivero | JohnIRivero@jourrapide.com | Quiz #1 | 95 |
| 2 | John I. Rivero | JohnIRivero@jourrapide.com | Quiz #2 | 78 |
| 3 | John I. Rivero | JohnIRivero@jourrapide.com | Midterm Exam | 82 |
| ... | ... | ... | ... | ... |

15 / 77

# Overview

## Deletion anomalies

Deletion anomalies occur when you delete a record, but because of the design of the tables, you accidentally delete information you shouldn't have.

16 / 77

# Overview

## Deletion anomalies

Deletion anomalies occur when you delete a record, but because of the design of the tables, you accidentally delete information you shouldn't have.

- For example, if `Helen C. Gonzalez` has only taken one assessment, Quiz #1, but we decide to drop that grade. Deleting that record would remove her email address entirely from our student data.

| id | name | email | assessment_title | grade |
|---|---|---|---|---|
| 4 | Mary G. Dickinson | MaryGDickinson@jourrapide.com | Quiz #1 | 95 |
| 5 | JSandra B. Kile | SandraBKile@teleworm.us | Quiz #2 | 78 |
| 6 | Helen C. Gonzalez | HelenCGonzalez@teleworm.us | Quiz #1 | 82 |
| ... | ... | ... | ... | ... |

17 / 77


First Normal Form
18 / 77

# First Normal Form

## Introduction

First normal form deals with the **shape** of a record type.

19 / 77

# First Normal Form

## Introduction

First normal form deals with the **shape** of a record type.

- All records in a table must contain *the same number of fields*.

# First Normal Form

## Introduction

First normal form deals with the **shape** of a record type.

- All records in a table must contain *the same number of fields*.
- In other words, all tables in relational database systems have a *fixed schema*.

# First Normal Form

## Introduction

First normal form deals with the **shape** of a record type.

- All records in a table must contain *the same number of fields*.
- In other words, all tables in relational database systems have a *fixed schema*.
- A fixed schema is generally a requirement of modern relational database systems, and requires no extra work.

# First Normal Form

## Introduction

First normal form deals with the **shape** of a record type.

- All records in a table must contain *the same number of fields*.
- In other words, all tables in relational database systems have a *fixed schema*.
- A fixed schema is generally a requirement of modern relational database systems, and requires no extra work.
- All values in a given field should also be *singular values*.

# Second Normal Form

# Second Normal Form

## Introduction

Second and third normal forms both deal with **the relationship between non-key and key fields**.

25 / 77

# Second Normal Form

## Introduction

Second and third normal forms both deal with **the relationship between non-key and key fields**.

- Each record in second and third normal forms *must satisfy first normal form*.

# Second Normal Form

## Introduction

Second and third normal forms both deal with **the relationship between non-key and key fields**.

- Each record in second and third normal forms *must satisfy first normal form*.
- A non-key field *must provide a fact about the entity uniquely identified by the primary key*.

27 / 77

# Second Normal Form

## Introduction

Second and third normal forms both deal with **the relationship between non-key and key fields**.

- Each record in second and third normal forms *must satisfy first normal form*.
- A non-key field *must provide a fact about the entity uniquely identified by the primary key*.
- It is not allowed for a non-key field to provide a fact about only a part of that entity or about some other unrelated entity.

28 / 77

# Second Normal Form

## Introduction

Second and third normal forms both deal with **the relationship between non-key and key fields**.

- Each record in second and third normal forms *must satisfy first normal form*.
- A non-key field *must provide a fact about the entity uniquely identified by the primary key*.
- It is not allowed for a non-key field to provide a fact about only a part of that entity or about some other unrelated entity.
- The fact could be a one-to-many relationship, such as the department of an employee, or a one-to-one relationship, such as the spouse of an employee.

29 / 77

# Second Normal Form

## Applicability

Second normal form *only applies to tables whose primary key is composed of two or more fields*.

30 / 77

# Second Normal Form

## Applicability

Second normal form *only applies to tables whose primary key is composed of two or more fields*.

```
CREATE TABLE (
    part TEXT NOT NULL
    warehouse TEXT NOT NULL
    quantity INTEGER
    warehouse-address TEXT
    PRIMARY KEY (part, warehouse)
)
```

31 / 77

# Second Normal Form

## Applicability

Second normal form *only applies to tables whose primary key is composed of two or more fields*.

```
CREATE TABLE (
    part TEXT NOT NULL
    warehouse TEXT NOT NULL
    quantity INTEGER
    warehouse-address TEXT
    PRIMARY KEY (part, warehouse)
)
```

- While today it is possible to create such **composite keys**, it is increasingly uncommon, in preference for singular **surrogate key** fields containing an auto-incrementing arbitrary integer.

# Second Normal Form

## Applicability

Second normal form *only applies to tables whose primary key is composed of two or more fields*.

```
CREATE TABLE (
    part TEXT NOT NULL
    warehouse TEXT NOT NULL
    quantity INTEGER
    warehouse-address TEXT
    PRIMARY KEY (part, warehouse)
)
```

- While today it is possible to create such **composite keys**, it is increasingly uncommon, in preference for singular **surrogate key** fields containing an auto-incrementing arbitrary integer.
- Nevertheless, we will explore it.

33 / 77

# Second Normal Form

## Example

Take, for example, the following table showing parts inventories in various warehouses:

34 / 77

# Second Normal Form

## Example

Take, for example, the following table showing parts inventories in various warehouses:

| part | warehouse | quantity | warehouse-address |
|---|---|---|---|
| Baby Bed Crib Screws Hardware Replacement Kit, cSeao 25-Set | Avenel, NJ | 2441 | 275 Omar Ave, Avenel, NJ 07001 |
| Prime-Line N 7534 Bi-Fold Door Hardware Repair Kit | Florence, NJ | 1121 | 309 Cedar Ln, Florence, NJ 08518 |
| HIMIKI Tailgate Hardware Rebuild Kit w/Handle Bezel Latch Cable | Avenel, NJ | 3567 | 275 Omar Ave, Avenel, NJ 07001 |
| ... | ... | ... | ... |

35 / 77

# Second Normal Form

## Example

Take, for example, the following table showing parts inventories in various warehouses:

| part | warehouse | quantity | warehouse-address |
|---|---|---|---|
| Baby Bed Crib Screws Hardware Replacement Kit, cSeao 25-Set | Avenel, NJ | 2441 | 275 Omar Ave, Avenel, NJ 07001 |
| Prime-Line N 7534 Bi-Fold Door Hardware Repair Kit | Florence, NJ | 1121 | 309 Cedar Ln, Florence, NJ 08518 |
| HIMIKI Tailgate Hardware Rebuild Kit w/Handle Bezel Latch Cable | Avenel, NJ | 3567 | 275 Omar Ave, Avenel, NJ 07001 |
| ... | ... | ... | ... |

The composite primary key is composed of **part** and **warehouse**, meaning that the combination of those two fields is guaranteed to be unique for each record.

36 / 77

# Second Normal Form

## Problems

This example does not meet the requirements of second normal form.

# Second Normal Form

## Problems

This example does not meet the requirements of second normal form.

- The field `warehouse-address` is a fact about the `warehouse` only, not a fact about the `part` / `warehouse` combined entity that this table is about.

38 / 77

# Second Normal Form

## Problems

This example does not meet the requirements of second normal form.

- The field `warehouse-address` is a fact about the `warehouse` only, not a fact about the `part` / `warehouse` combined entity that this table is about.
- This is unfortunate, since it requires the address of each warehouse to be repeated however many times there are parts in that warehouse.

39 / 77

# Second Normal Form

## Problems

This example does not meet the requirements of second normal form.

- The field `warehouse-address` is a fact about the `warehouse` only, not a fact about the `part` / `warehouse` combined entity that this table is about.
- This is unfortunate, since it requires the address of each warehouse to be repeated however many times there are parts in that warehouse.
- This **data redundancy** makes maintaining the data difficult. Updating a warehouses address would have to be done across many records, not just in a singular place.

40 / 77

# Second Normal Form

## Problems

This example does not meet the requirements of second normal form.

- The field `warehouse-address` is a fact about the `warehouse` only, not a fact about the `part` / `warehouse` combined entity that this table is about.
- This is unfortunate, since it requires the address of each warehouse to be repeated however many times there are parts in that warehouse.
- This **data redundancy** makes maintaining the data difficult. Updating a warehouses address would have to be done across many records, not just in a singular place.
- If there were no parts stored in a given warehouse, there would be nowhere in the database to store the address of that warehouse.

41 / 77

# Second Normal Form

## Solutions

In order to normalize these records, we could easily split the data into two tables.

# Second Normal Form

## Solutions

In order to normalize these records, we could easily split the data into two tables.

One table for `parts`:

| part | warehouse_id | quantity |
|---|---|---|
| Baby Bed Crib Screws Hardware Replacement Kit, cSeao 25-Set | 1 | 2441 |
| Prime-Line N 7534 Bi-Fold Door Hardware Repair Kit | 2 | 1121 |
| HIMIKI Tailgate Hardware Rebuild Kit w/Handle Bezel Latch Cable | 1 | 3567 |
| ... | ... | ... |

43 / 77

# Second Normal Form

## Solutions

In order to normalize these records, we could easily split the data into two tables.

One table for `parts`:

| part | warehouse_id | quantity |
|---|---|---|
| Baby Bed Crib Screws Hardware Replacement Kit, cSeao 25-Set | 1 | 2441 |
| Prime-Line N 7534 Bi-Fold Door Hardware Repair Kit | 2 | 1121 |
| HIMIKI Tailgate Hardware Rebuild Kit w/Handle Bezel Latch Cable | 1 | 3567 |
| ... | ... | ... |

And another for `warehouses`:

| id | address |
|---|---|
| 1 | 275 Omar Ave, Avenel, NJ 07001 |
| 2 | 309 Cedar Ln, Florence, NJ 08518 |
| ... | ... |

# Third Normal Form

45 / 77

# Third Normal Form

## Introduction

As with second normal form, third normal forms deals with **the relationship between non-key and key fields**.

46 / 77

# Third Normal Form

## Introduction

As with second normal form, third normal forms deals with **the relationship between non-key and key fields**.

- Records in third normal form must *satisfy second normal form*.

47 / 77

# Third Normal Form

## Introduction

As with second normal form, third normal forms deals with **the relationship between non-key and key fields**.

- Records in third normal form must *satisfy second normal form*.
- Third normal form is *violated when a non-key field is a fact about another non-key field*.

48 / 77

# Third Normal Form

## Introduction

As with second normal form, third normal forms deals with **the relationship between non-key and key fields**.

- Records in third normal form must *satisfy second normal form*.
- Third normal form is *violated when a non-key field is a fact about another non-key field*.
- Whereas second normal form is only applicable to tables with composite primary keys, third normal form applies to all tables.

49 / 77

# Third Normal Form

## Example

Take, for example, a table about `employees` at a company.

| id | employee | department | location |
|---|---|---|---|
| 1 | Henry K. Brinkman | Accounting | Fort Myers, FL |
| 2 | Darlene R. Gonzalez | Marketing | Jackson, MS |
| 3 | Abigail W. Wagner | Sales | Pleasantville, NJ |
| ... | ... | ... | ... |

50 / 77

# Third Normal Form

## Example

Take, for example, a table about `employees` at a company.

| id | employee | department | location |
|---|---|---|---|
| 1 | Henry K. Brinkman | Accounting | Fort Myers, FL |
| 2 | Darlene R. Gonzalez | Marketing | Jackson, MS |
| 3 | Abigail W. Wagner | Sales | Pleasantville, NJ |
| ... | ... | ... | ... |

- Let's assume that the `location` is dependent upon the `department`, with each department having a different location.

51 / 77

# Third Normal Form

## Problems

This example does not meet the requirements of third normal form.

# Third Normal Form

## Problems

This example does not meet the requirements of third normal form.

- The `id` field is the primary key and represents an employee. If each department is located in one place, then the `location` field is a fact about the `department`, and not a fact about the employee.

53 / 77

# Third Normal Form

## Problems

This example does not meet the requirements of third normal form.

- The `id` field is the primary key and represents an employee. If each department is located in one place, then the `location` field is a fact about the `department`, and not a fact about the employee.
- The department's location is repeated in the record of every employee assigned to that department. If the location of the department changes, every such record must be updated.

54 / 77

# Third Normal Form

## Problems

This example does not meet the requirements of third normal form.

- The `id` field is the primary key and represents an employee. If each department is located in one place, then the `location` field is a fact about the `department`, and not a fact about the employee.
- The department's location is repeated in the record of every employee assigned to that department. If the location of the department changes, every such record must be updated.
- Because of the **redundancy**, the data might become inconsistent, with different records showing different locations for the same department.

55 / 77

# Third Normal Form

## Problems

This example does not meet the requirements of third normal form.

- The `id` field is the primary key and represents an employee. If each department is located in one place, then the `location` field is a fact about the `department`, and not a fact about the employee.
- The department's location is repeated in the record of every employee assigned to that department. If the location of the department changes, every such record must be updated.
- Because of the **redundancy**, the data might become inconsistent, with different records showing different locations for the same department.
- If a department has no employees, there may be no record in which to keep the department's location.

56 / 77

# Third Normal Form

## Solution

As with violations of second normal form, the solution to a violation of third normal form is typically to split the data into multiple tables.

57 / 77

# Third Normal Form

## Solution

As with violations of second normal form, the solution to a violation of third normal form is typically to split the data into multiple tables.

One table for `employees`:

| id | employee | department_id |
|---|---|---|
| 1 | Henry K. Brinkman | 1 |
| 2 | Darlene R. Gonzalez | 2 |
| 3 | Abigail W. Wagner | 3 |
| ... | ... | ... |

58 / 77

# Third Normal Form

## Solution

As with violations of second normal form, the solution to a violation of third normal form is typically to split the data into multiple tables.

One table for `employees`:

| id | employee | department_id |
|---|---|---|
| 1 | Henry K. Brinkman | 1 |
| 2 | Darlene R. Gonzalez | 2 |
| 3 | Abigail W. Wagner | 3 |
| ... | ... | ... |

And another for `departments`:

| id | department | location |
|---|---|---|
| 1 | Accounting | Fort Myers, FL |
| 2 | Marketing | Jackson, MS |
| 3 | Sales | Pleasantville, NJ |
| ... | ... | ... |

59 / 77

# Fourth Normal Form

60 / 77

# Fourth Normal Form

## Introduction

Fourth normal form is concerned with multi-valued facts, which we'll show by example. In order for a record to meet fourth normal form, it must:

61 / 77

# Fourth Normal Form

## Introduction

Fourth normal form is concerned with multi-valued facts, which we'll show by example. In order for a record to meet fourth normal form, it must:

- satisfy the requirements of third normal form.

62 / 77

# Fourth Normal Form

## Introduction

Fourth normal form is concerned with multi-valued facts, which we'll show by example. In order for a record to meet fourth normal form, it must:

- satisfy the requirements of third normal form.
- **not** contain more than one independent **multi-valued fact** about an entity.

63 / 77

# Fourth Normal Form

## Example

For example, consider a situation where we intend to store employee's skills and foreign language abilities.

# Fourth Normal Form

## Example

For example, consider a situation where we intend to store employee's skills and foreign language abilities.

A single employee who has multiple skills and/or multiple languages (two independent multi-valued facts about them) might [erroneously] be represented with two or more independent multi-valued fact fields.

65 / 77

# Fourth Normal Form

## Example

For example, consider a situation where we intend to store employee's skills and foreign language abilities.

A single employee who has multiple skills and/or multiple languages (two independent multi-valued facts about them) might [erroneously] be represented with two or more independent multi-valued fact fields.

| id | employee | skill | language |
|---|---|---|---|
| 1 | Henry K. Brinkman | cook | |
| 2 | Henry K. Brinkman | type | |
| 3 | Henry K. Brinkman | | French |
| 4 | Henry K. Brinkman | | German |
| 5 | Henry K. Brinkman | | Greek |
| ... | ... | ... | ... |

66 / 77

# Fourth Normal Form

## Example

For example, consider a situation where we intend to store employee's skills and foreign language abilities.

A single employee who has multiple skills and/or multiple languages (two independent multi-valued facts about them) might [erroneously] be represented with two or more independent multi-valued fact fields.

| id | employee | skill | language |
|---|---|---|---|
| 1 | Henry K. Brinkman | cook | |
| 2 | Henry K. Brinkman | type | |
| 3 | Henry K. Brinkman | | French |
| 4 | Henry K. Brinkman | | German |
| 5 | Henry K. Brinkman | | Greek |
| ... | ... | ... | ... |

- In this representation, in addition to **data redundancy**, there is **ambiguity** in the meaning of the null values - does the employee lack those abilities, are they not applicable, or are they unknown?

67 / 77

# Fourth Normal Form

## Another Possibility

That same data, with two or more multi-value fact fields, might be represented a few different ways, including:

| id | employee | skill | language |
|---|---|---|---|
| 1 | Henry K. Brinkman | cook | French |
| 2 | Henry K. Brinkman | type | German |
| 3 | Henry K. Brinkman | type | Greek |
| ... | ... | ... | ... |

# Fourth Normal Form

## Another Possibility

That same data, with two or more multi-value fact fields, might be represented a few different ways, including:

| id | employee | skill | language |
|---|---|---|---|
| 1 | Henry K. Brinkman | cook | French |
| 2 | Henry K. Brinkman | type | German |
| 3 | Henry K. Brinkman | type | Greek |
| ... | ... | ... | ... |

- In this representation, we have removed the null values, but we still have **redundancy** and therefore difficulty maintaining data.

69 / 77

# Fourth Normal Form

## Another Possibility

That same data, with two or more multi-value fact fields, might be represented a few different ways, including:

| id | employee | skill | language |
|---|---|---|---|
| 1 | Henry K. Brinkman | cook | French |
| 2 | Henry K. Brinkman | type | German |
| 3 | Henry K. Brinkman | type | Greek |
| ... | ... | ... | ... |

- In this representation, we have removed the null values, but we still have **redundancy** and therefore difficulty maintaining data.
- Note that the `skill` and `language` fields are said in our description of the data to be **independent**. Thus, this model is forbidden by the fourth normal form.

70 / 77

# Fourth Normal Form

## Another Possibility

That same data, with two or more multi-value fact fields, might be represented a few different ways, including:

| id | employee | skill | language |
|---|---|---|---|
| 1 | Henry K. Brinkman | cook | French |
| 2 | Henry K. Brinkman | type | German |
| 3 | Henry K. Brinkman | type | Greek |
| ... | ... | ... | ... |

- In this representation, we have removed the null values, but we still have **redundancy** and therefore difficulty maintaining data.
- Note that the `skill` and `language` fields are said in our description of the data to be **independent**. Thus, this model is forbidden by the fourth normal form.
- However, if a skill was dependent upon a specific language, this model would be allowed by the fourth normal form.

71 / 77

# Fourth Normal Form

## Solution

The solution to remove redundancy, ambiguity, and anomalies, as with previous normal forms, is to split the data up into multiple tables.

72 / 77

# Fourth Normal Form

## Solution

The solution to remove redundancy, ambiguity, and anomalies, as with previous normal forms, is to split the data up into multiple tables.

One for `employees`:

| id | employee |
|---|---|
| 1 | Henry K. Brinkman |
| 2 | Darlene R. Gonzalez |
| 3 | Abigail W. Wagner |
| ... | ... |

# Fourth Normal Form

## Solution (continued)

The solution, as with previous normal forms, is to split the data up into multiple tables.

Another for `skills`:

| id | employee_id | skill |
|---|---|---|
| 1 | 1 | cook |
| 2 | 1 | type |
| 3 | 1 | type |
| ... | ... | ... |

74 / 77

# Fourth Normal Form

## Solution (continued again)

The solution, as with previous normal forms, is to split the data up into multiple tables.

And a third for `languages`:

| id | employee_id | language |
|---|---|---|
| 1 | 1 | French |
| 2 | 1 | German |
| 3 | 1 | Greek |
| ... | ... | ... |

# Conclusions

76 / 77

# Conclusions

Thank you. Bye.

77 / 77