Assignment5No2

Unnormalized Table & Part A

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| StudentID | Student Name | Address | Email | Classes | Major |
| S1 | Joe Green | 124 Main St. | [Joe@school.edu](mailto:Joe@school.edu) | IT1025/MATH1200  IT1050 | Programming |
| S2 | Sue Smith | 345 Second St. | [Sue@school.edu](mailto:Sue@school.edu) | IT1025/IT1050/IT2351 | Programming |
| S3 | Nick Green | 45 York Road | [Nick@school.edu](mailto:Nick@school.edu) | 1025 | Networking |
| S4 | Andy Andrews | 600 5th Ave. | [Andy@school.edu](mailto:Andy@school.edu) | 1025/1050 | Networking |
| S5 | Buck Major | 987 Elm Blvd. | [Buck@school.edu](mailto:Buck@school.edu) | MATH1200 | Programming |
| S6 | Sally Smith | 67 Broad Ln. | [Sally@school.edu](mailto:Sally@school.edu) | 1050 | Networking |
| S7 | Bo Fletcher | 5 Sugar Cir. | [Bo@school.com](mailto:Bo@school.com) | IT1025/IT1050 | Programming |

Part C

1st Normal Form: No repeating groups

Technically Classes is a repeating group. I split those up.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| StudentID | Student Name | Address | Email | Classes | Major |
| S1 | Joe Green | 124 Main St. | [Joe@school.edu](mailto:Joe@school.edu) | IT1025 | Programming |
| S1 | Joe Green | 124 Main St. | [Joe@school.edu](mailto:Joe@school.edu) | MATH1200 | Programming |
| S1 | Joe Green | 124 Main St. | [Joe@school.edu](mailto:Joe@school.edu) | IT1050 | Programming |
| S2 | Sue Smith | 345 Second St. | [Sue@school.edu](mailto:Sue@school.edu) | IT1025 | Programming |
| S2 | Sue Smith | 345 Second St. | [Sue@school.edu](mailto:Sue@school.edu) | IT1050 | Programming |
| S2 | Sue Smith | 345 Second St. | [Sue@school.edu](mailto:Sue@school.edu) | IT2351 | Programming |
| S3 | Nick Green | 45 York Road | [Nick@school.edu](mailto:Nick@school.edu) | 1025 | Networking |
| S4 | Andy Andrews | 600 5th Ave. | [Andy@school.edu](mailto:Andy@school.edu) | 1025 | Networking |
| S4 | Andy Andrews | 600 5th Ave. | [Andy@school.edu](mailto:Andy@school.edu) | 1050 | Networking |
| S5 | Buck Major | 987 Elm Blvd. | [Buck@school.edu](mailto:Buck@school.edu) | MATH1200 | Programming |
| S6 | Sally Smith | 67 Broad Ln. | [Sally@school.edu](mailto:Sally@school.edu) | 1050 | Networking |
| S7 | Bo Fletcher | 5 Sugar Cir. | [Bo@school.com](mailto:Bo@school.com) | IT1025 | Programming |
| S7 | Bo Fletcher | 5 Sugar Cir. | [Bo@school.com](mailto:Bo@school.com) | IT1050 | Programming |

2nd Normal Form: Eliminate Redundant Data

A-Classes and Major Become their own table Retaining StudentID

School\_Studies

|  |  |  |
| --- | --- | --- |
| StudentID | Classes | Major |
| S1 | IT1025 | Programming |
| S1 | MATH1200 | Programming |
| S1 | IT1050 | Programming |
| S2 | IT1025 | Programming |
| S2 | IT1050 | Programming |
| S2 | IT2351 | Programming |
| S3 | 1025 | Networking |
| S4 | 1025 | Networking |
| S4 | 1050 | Networking |
| S5 | MATH1200 | Programming |
| S6 | 1050 | Networking |
| S7 | IT1025 | Programming |
| S7 | IT1050 | Programming |

B-Names Get Split Up.

Student\_Names

|  |  |  |
| --- | --- | --- |
| StudentID | First\_Name | Last\_Name |
| S1 | Joe | Green |
| S2 | Sue | Smith |
| S3 | Nick | Green |
| S4 | Andy | Andrews |
| S5 | Buck | Major |
| S6 | Sally | Smith |
| S7 | Bo | Fletcher |

3rd Normal Form: Eliminate data not dependent on key

Address and Email get their own table.

Communication\_Forms

|  |  |  |
| --- | --- | --- |
| StudentID | Address | Email |
| S1 | 124 Main St. | [Joe@school.edu](mailto:Joe@school.edu) |
| S2 | 345 Second St. | [Sue@school.edu](mailto:Sue@school.edu) |
| S3 | 45 York Road | [Nick@school.edu](mailto:Nick@school.edu) |
| S4 | 600 5th Ave. | [Andy@school.edu](mailto:Andy@school.edu) |
| S5 | 987 Elm Blvd. | [Buck@school.edu](mailto:Buck@school.edu) |
| S6 | 67 Broad Ln. | [Sally@school.edu](mailto:Sally@school.edu) |
| S7 | 5 Sugar Cir. | [Bo@school.com](mailto:Bo@school.com) |

Part B

**Student\_Names** gets the Primary Key Stud\_Num

StudentID becomes a reference for School\_Studies & Comm\_types

|  |  |  |  |
| --- | --- | --- | --- |
| Stud\_Num\* | StudentID\*\* | First\_Name | Last\_Name |
| 1 | S1 | Joe | Green |
| 2 | S2 | Sue | Smith |
| 3 | S3 | Nick | Green |
| 3 | S4 | Andy | Andrews |
| 4 | S5 | Buck | Major |
| 5 | S6 | Sally | Smith |
| 6 | S7 | Bo | Fletcher |

StudentID: Becomes Primary Key for School\_Studies & Comm\_types

**Comm\_types**

|  |  |  |
| --- | --- | --- |
| StudentID\* | Address | Email |
| S1 | 124 Main St. | [Joe@school.edu](mailto:Joe@school.edu) |
| S2 | 345 Second St. | [Sue@school.edu](mailto:Sue@school.edu) |
| S3 | 45 York Road | [Nick@school.edu](mailto:Nick@school.edu) |
| S4 | 600 5th Ave. | [Andy@school.edu](mailto:Andy@school.edu) |
| S5 | 987 Elm Blvd. | [Buck@school.edu](mailto:Buck@school.edu) |
| S6 | 67 Broad Ln. | [Sally@school.edu](mailto:Sally@school.edu) |
| S7 | 5 Sugar Cir. | [Bo@school.com](mailto:Bo@school.com) |

**School\_Studies**

|  |  |  |
| --- | --- | --- |
| StudentID\* | Classes | Major |
| S1 | IT1025 | Programming |
| S1 | MATH1200 | Programming |
| S1 | IT1050 | Programming |
| S2 | IT1025 | Programming |
| S2 | IT1050 | Programming |
| S2 | IT2351 | Programming |
| S3 | 1025 | Networking |
| S4 | 1025 | Networking |
| S4 | 1050 | Networking |
| S5 | MATH1200 | Programming |
| S6 | 1050 | Networking |
| S7 | IT1025 | Programming |
| S7 | IT1050 | Programming |

Assignment5No2 Proposed SQL

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- Assignment5No2 IT2351 Jennifer Howard

-- this SQL script creates a database for the provided student info

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- Create the database

Drop database if exists student\_log;

Create database student\_log;

-- select the databse

use student\_log;

-- create the tables

create table student\_names

(

stud\_num int primary key auto\_increment,

studentID varchar(5) not null unique,

first\_name varchar(50),

last\_name varchar(50)

);

create table comm\_types

(

studentID varchar(5) primary key,

address varchar(50),

email varchar(50),

constraint comm\_types\_fk\_student\_names

foreign key (studentID) references student\_names (studentID)

);

-- create table school\_studies

(

studentID varchar(5) primary key,

classes varchar(15),

major varchar(50),

constraint school\_studies\_fk\_student\_names

foreign key (studentID) references student\_names (studentID)

);

Assignment5No3 Proposed SQL

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- Assignment5No3 IT2351 Jenn Howard

-- this SQL script creates a view named items\_ordered using the guitar database

-- that shows the item\_name, product name, item price,and quantity for each item ordered

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Use jenns\_guitar\_database;

Create view items\_ordered as

Select item\_id as item\_name, product\_id as product\_name,

item\_price, quantity

From order\_items

Order by item\_name, product\_name, item\_price, quantity;