Database Design

Wednesday, January 25, 2017

Agenda

- Announcements

- Reading Quiz

- Conceptual Diagram

- Practice Problems 1-2

- Physical Diagram

- Practice Problem 3

Announcements

- Reminder: send email about your team by tonight:

    - Your full names, EIDs, Github usernames

    - Team name

    - Use email subject line: CS327E Team XYZ, where XYZ is your team name

    - Send email to me and both TAs, copy your partner on the email

- Lab 1 setup guide: https://github.com/wolfier/CS327E/wiki

- Next week: Lab 1

Q1: Do you have to follow Codd's rule for Relational DBMS?

a) Yes, because you'll get sent to jail

b) No, but there are consequences

Q2: What is/are required for all SQL commands?

a) Keywords

b) Tables

c) Columns

d) Both a and c

e) Both a and b

Q3: What are small programs that are built into the SQL? For example, AVG.

a) Tables

b) Functions

c) Columns

d) Keywords

Q4: Some of the native data type categories for PSQL is/are...

a) Numeric type

b) Character type

c) Date and time type

d) All of the above

Q5: "CREATE TABLE" SQL statement cannot

a) Clone a table

b) Materialize the result of the SELECT

c) Delete a table

Recall: Key Concepts

- Entity = an object of interest

- Attribute = property of an Entity

- Relationship = association between two Entities

- Relationship types:

  one-to-one: |———————|

  one-to-many: |———————<   many-to-one: >———————|

  many-to-many: >———————<

Scenario: SXSW Database

Design a database for the organizers of the music festival to help them gain more insight into their current customer base. Want to use this database answer questions such as **which shows were well-attended last year? Who were the most popular artists and groups/bands based on number of signups? Which customers are loyal and return to the festival year-after-year? Which customers are new and which ones only go to free concerts? Which customers are branching out and sign up for other types of events (networking, etc.)?** These are just a few questions, I'm sure you can think of others to help the organizers of the festival :))

Customers

- cust_num
- first_name
- last_name
- street
- city
- state
- zip_code
- country

Artists

- artist_num
- first_name
- last_name
- instrument
- group
- birth_year

Groups

- group_num
- group_name
- genre
- home_town
- home_state
- home_country

Signups

- cust_num
- show_num
- signup_date

Shows

- show_num
- venue
- sponsor
- show_date
- show_start_time
- ticket_price
- total_ticket_count
- remaining_tickets

Practice Problem 1: What type of relationship exists between Shows and Groups?

a) One-to-one

b) One-to-many

c) Many-to-one

d) Many-to-many

e) No relationship

Practice Problem 2: Improve this design by allowing Groups to have multiple genres

Customers
- cust_num
- first_name
- last_name
- street
- city
- state
- zip_code
- country

Artists
- artist_num
- first_name
- last_name
- instrument
- group
- birth_year

Groups
- group_num
- group_name
- genre
- home_town
- home_state
- home_country

Signups
- cust_num
- show_num
- signup_date

Shows
- show_num
- venue
- sponsor
- show_date
- show_start_time
- ticket_price
- total_ticket_count
- remaining_tickets
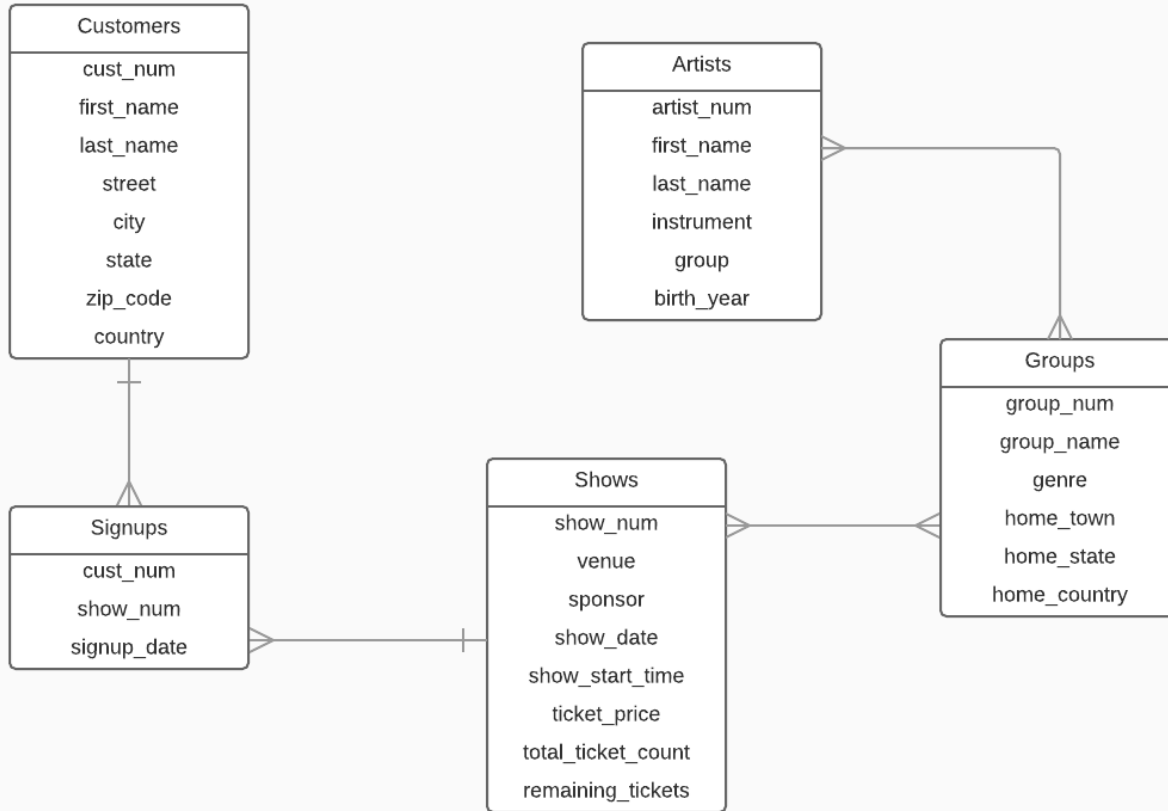
Practice Problem 2: How many Entities does your new diagram have?

a) 4

b) 5

c) 6

d) 7

Converting to Physical Diagram

Conceptual diagram represents:

- Entities, attributes, relationship types

Conceptual diagram doesn't represent:

- Keys, junction tables, datatypes

Converting *m:n* Relationship Types
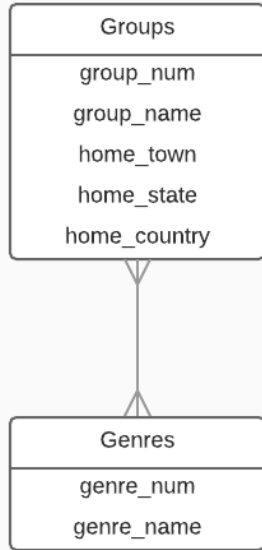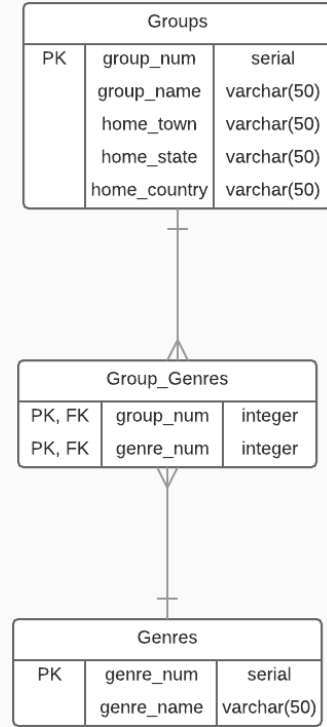
Conceptual diagram

Physical diagram

Customers
- cust_num
- first_name
- last_name
- street
- city
- state
- zip_code
- country

Artists
- artist_num
- first_name
- last_name
- instrument
- group
- birth_year

Groups
- group_num
- group_name
- home_town
- home_state
- home_country

Shows
- show_num
- venue
- sponsor
- show_date
- show_start_time
- ticket_price
- total_ticket_count
- remaining_tickets

Signups
- cust_num
- show_num
- signup_date

Genres
- genre_num
- genre_name
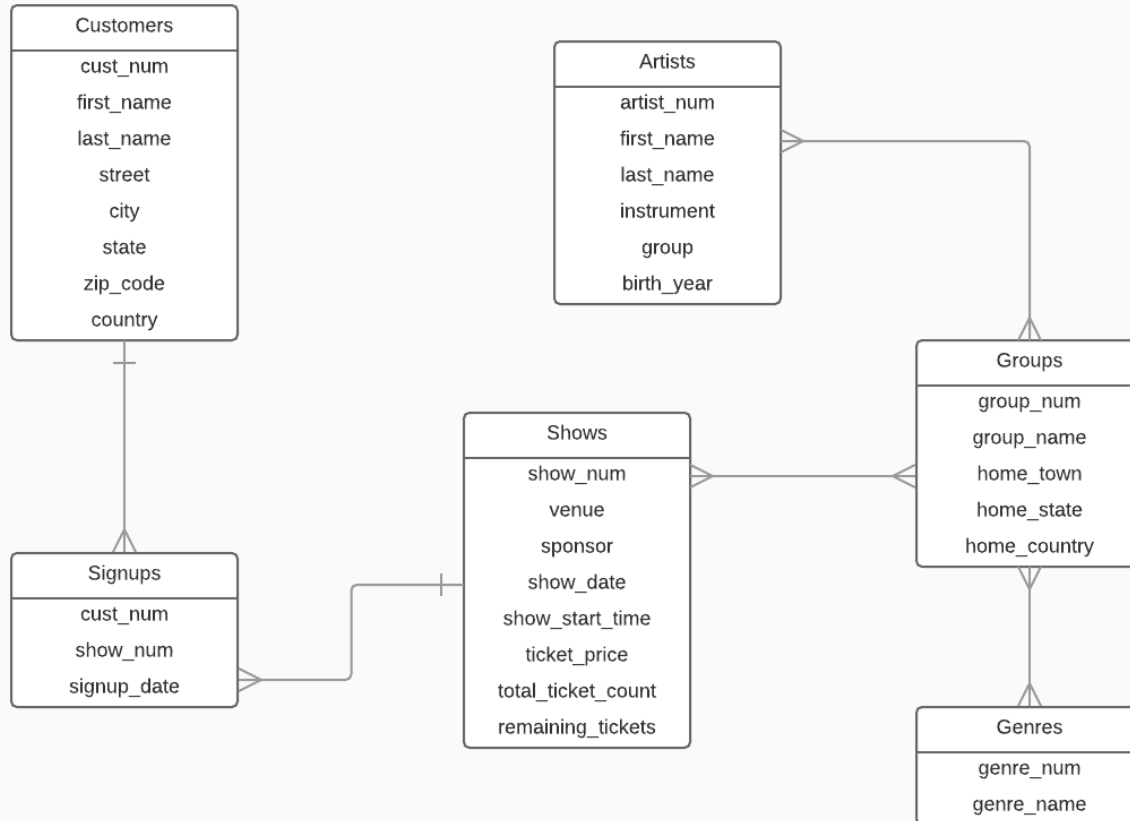
Practice Problem 3: How many junction tables does your physical diagram have?

a) 0

b) 1

c) 2

d) 3

SXSW Physical Diagram

Customers

PK	cust_num	serial
	first_name	varchar(50)
	last_name	varchar(50)
	street	varchar(50)
	city	varchar(50)
	state	char(2)
	zip_code	char(5)
	country	varchar(50)

Artists

PK	artist_num	serial
	first_name	varchar(50)
	last_name	varchar(50)
	birth_year	char(4)

Artists_Groups

PK, FK	artist_num	integer
PK, FK	group_num	integer
	instrument	varchar(50)

Groups

PK	group_num	serial
	group_name	varchar(50)
	home_town	varchar(50)
	home_state	varchar(50)
	home_country	varchar(50)

Acts

PK, FK	show_num	integer
PK, FK	group_num	integer
	act_type	char(1)

Signups

PK, FK	cust_num	integer
PK, FK	show_num	integer
	signup_date	timestamp

Shows

PK	show_num	serial
	venue	varchar(50)
	sponsor	varchar(50)
	show_date	date
	show_start_time	time
	ticket_price	decimal(5, 2)
	total_ticket_count	integer
	remaining_tickets	integer

Groups_Genres

PK, FK	group_num	integer
PK, FK	genre_num	integer

Genres

PK	genre_num	serial
	genre_name	varchar(50)