# Exam 1 Review

## Lecture 6

# Format

- 4 problems, with multiple sub-parts
- No notes, calculators, books, computers, phones, etc. may be used

# Content

Everything, including…

- General database knowledge
- The relational model
- SQL [programming]

# General Database Knowledge

- What is a transaction?
  - What are the properties that should hold for effective transaction processing?
- What is SQL?
  - DDL, DML

# Relational Model

- A database is composed of?
- A table schema is composed of?
- Each [schema component] has a ____ of valid _____ values?
- What is the difference between a set vs. bag of tuples?
  - In what context does each apply?
- Provide meaning/examples of each general category of constraints:
  - Implicit, Explicit, Application-based, Data Dependencies
- What kinds of constraints that can be defined in the schema?
  - What is a superkey vs. a key?
    - How do you identify a primary key? What happens to other super keys?
  - How do foreign keys fit in?

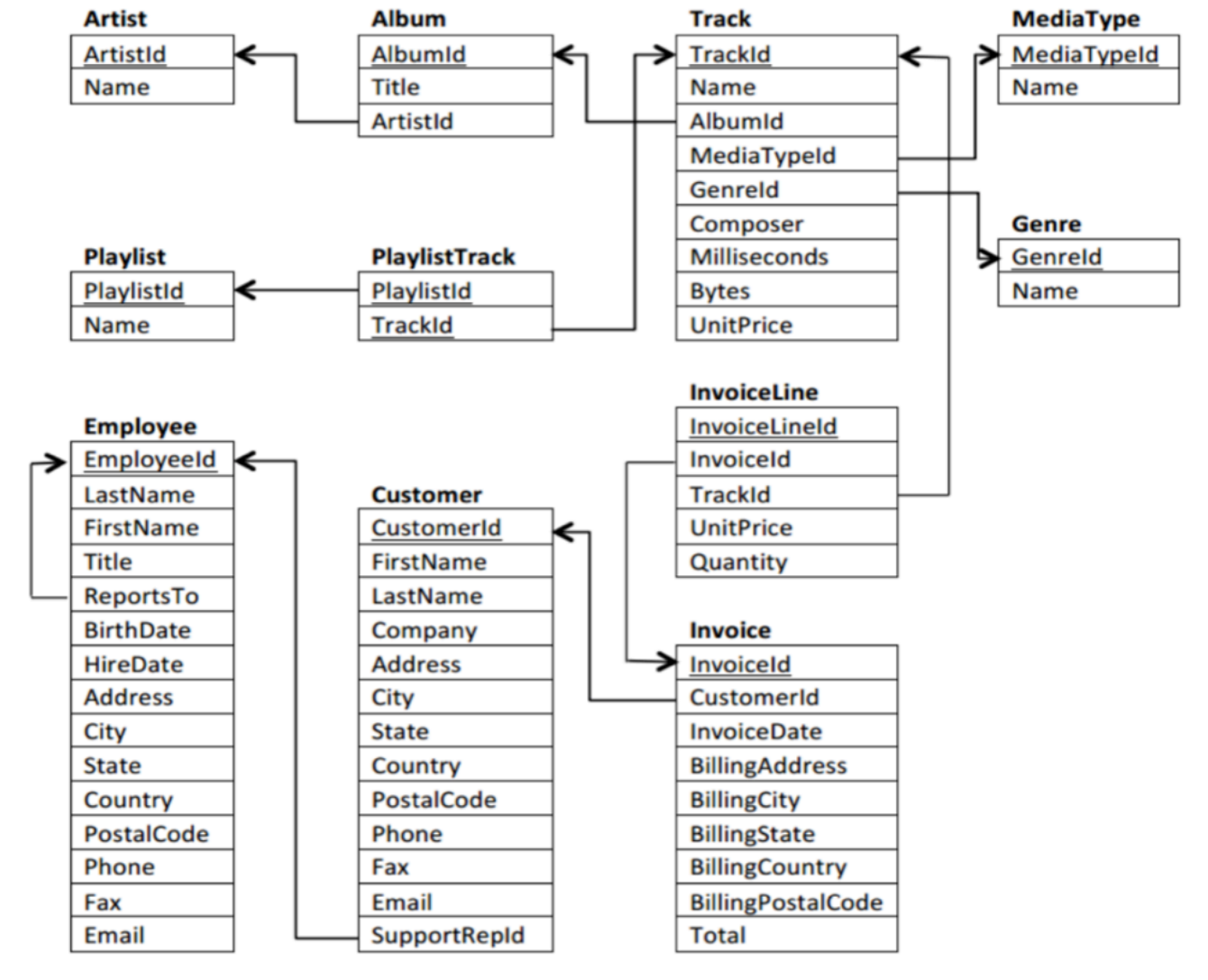
# SQL [Programming]

- Know your terms/keywords: DDL/DML, `ASSERTION`/`TRIGGER`, `BEGIN`/`COMMIT`/`ROLLBACK`, `GRANT`/`REVOKE`/`WITH GRANT OPTION`, `REFERENCES`, `CLOB`/`BLOB`, ODBC, Impedance Mismatch, Result Set, Cursor, SQL Injection attack

- Given a schema and a query description, write SQL [to create, modify, query]

- Given SQL and a set of populated table(s), predict the result set

# Chinook


Artist
ArtistId
Name
Album
AlbumId
Title
ArtistId
Track
TrackId
Name
AlbumId
MediaTypeId
GenreId
Composer
Milliseconds
Bytes
UnitPrice
MediaType
MediaTypeId
Name
Genre
GenreId
Name
Playlist
PlaylistId
Name
PlaylistTrack
PlaylistId
TrackId
InvoiceLine
InvoiceLineId
InvoiceId
TrackId
UnitPrice
Quantity
Employee
EmployeeId
LastName
FirstName
Title
ReportsTo
BirthDate
HireDate
Address
City
State
Country
PostalCode
Phone
Fax
Email
Customer
CustomerId
FirstName
LastName
Company
Address
City
State
Country
PostalCode
Phone
Fax
Email
SupportRepId
Invoice
InvoiceId
CustomerId
InvoiceDate
BillingAddress
BillingCity
BillingState
BillingCountry
BillingPostalCode
Total

# SQL Query (1)

For each genre, find the average length (in minutes) of each track in the "90's Music" playlist. Sort the results by length (longest first), then genre name (alphabetical). Only include those genres for which the average length is greater than 5 minutes.

| | g_name | avg_len_m |
|---|---|---|
| 1 | Electronica/Dance | 6.0114 |
| 2 | Alternative | 5.528175 |
| 3 | Classical | 5.32899333333333 |
| 4 | Jazz | 5.27926933333333 |
| 5 | Metal | 5.15945721544715 |

# Answer

```
SELECT g.Name AS g_name, AVG(t.milliseconds)/60000 AS avg_len_m
FROM ((track t INNER JOIN genre g ON t.GenreId=g.GenreId)
INNER JOIN PlaylistTrack pt ON t.TrackId=pt.TrackId)
INNER JOIN Playlist p ON pt.PlaylistId=p.PlaylistId
WHERE p.Name='90's Music'
GROUP BY g_name
HAVING AVG(t.milliseconds)/60000>5
ORDER BY avg_len_m DESC, g_name ASC;
```

# SQL Query (2)

**uno**

| a | b |
|---|---|
| 1 | foo |
| 2 | bar |
| 3 | baz |
| 4 | qux |

**dos**

| x | y |
|---|---|
| 2 | 1 |
| 3 | 1 |
| 7 | 4 |
| 8 | 3 |

1. Write the DDL and DML code to produce this database

1. Predict the outcome of the following query:

```
SELECT b AS var, AVG(x) AS avg_line
FROM uno LEFT JOIN dos ON uno.a=dos.y
GROUP BY b
ORDER BY b DESC;
```

# Answer (1)

```
CREATE TABLE uno (
a INT PRIMARY KEY,
b VARCHAR(10)
);

CREATE TABLE dos(
x INT,
y INT,
PRIMARY KEY (x, y),
CONSTRAINT y_fk FOREIGN KEY (y) REFERENCES uno (a)
);

INSERT INTO uno (a,b) VALUES (1,'foo');
INSERT INTO uno (a,b) VALUES (2,'bar');
INSERT INTO uno (a,b) VALUES (3,'baz');
INSERT INTO uno (a,b) VALUES (4,'qux');

INSERT INTO dos (x,y) VALUES (2,1);
INSERT INTO dos (x,y) VALUES (3,1);
INSERT INTO dos (x,y) VALUES (7,4);
INSERT INTO dos (x,y) VALUES (8,3);
```

# Answer (2)

| var | avg_line |
|---|---|
| qux | 7.0 |
| foo | 2.5 |
| baz | 8.0 |
| bar | *NULL* |