

Exam 1 Review

Lecture 7



Format & Process

- 1 hour (during class, Thursday 2/6)
 - Your time will start shortly after class begins
- 4 parts
 - General DBMS Knowledge: fill in words
 - Relational Model: matching, T/F, calculate
 - SQL (DDL/DML): SQL->info, what's wrong, result; Chinook->SQL
 - Security: T/F, explain
- Free to use: 1 front/back 8x11"
 - I will supply reference sheet, extra blank paper
 - Any violation of academic integrity will result in failing the class and a report to the University
- I will be in-person
 - And we'll debrief shortly after it's done :)



Content

- The Relational Model
- SQL
- Security



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?
- 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?
 - Practice with the DOCTOR
 - How do foreign keys fit in?
- What is a transaction? What are the ACID guarantees?



SQL

- Know terms/ideas of DDL/DML
 - Review “Create a DB from Scratch”
- Given a schema (Chinook), query description & result -> write SQL
 - Will avoid numerical/rounding issues and functions like PRINTF/IFNULL
- Given SQL and a set of table(s) -> predict the result

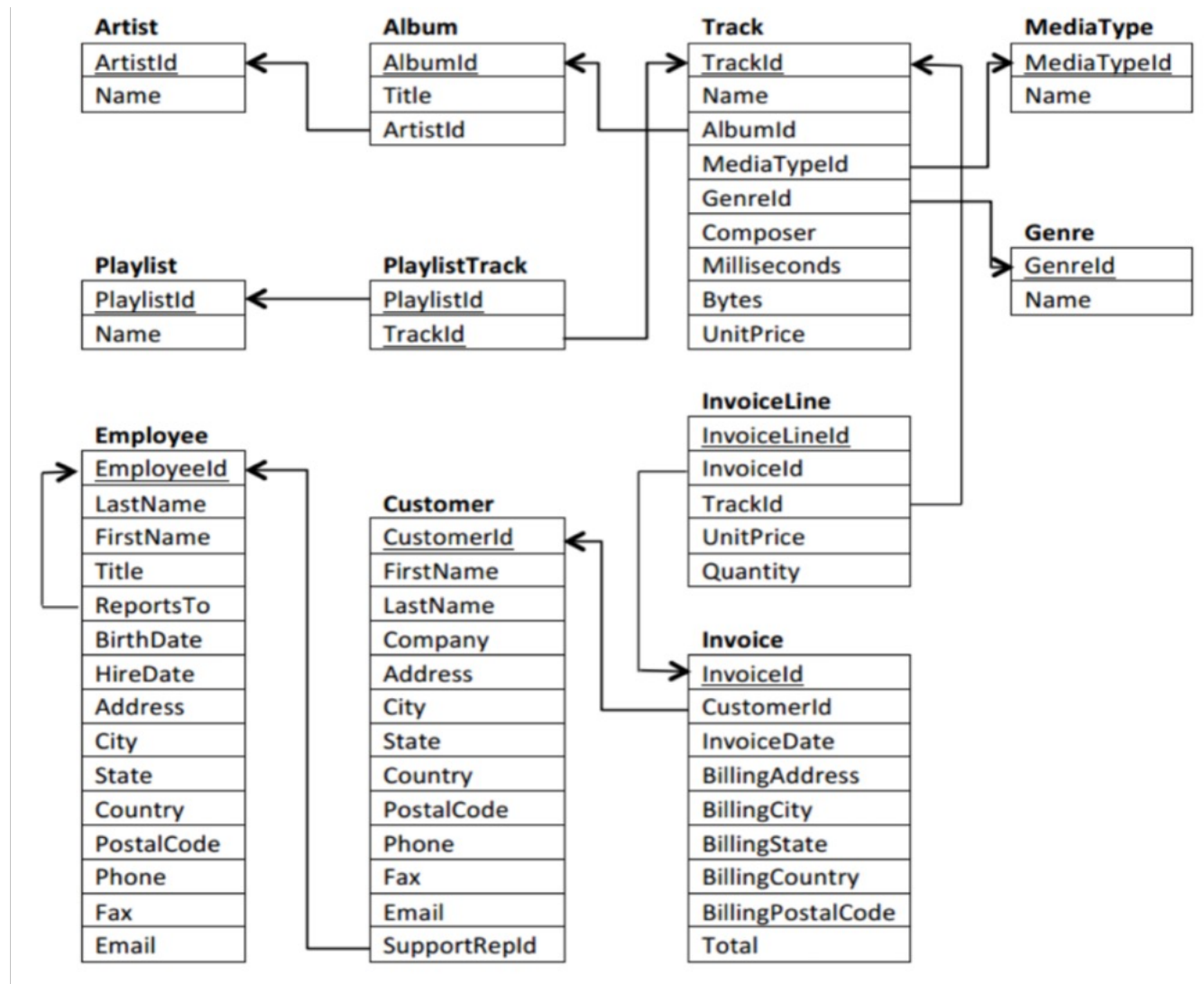


Security

- SQL Injection: meaning, prevention
- Secure password storage
- Methods of mitigating/detecting breaches



Chinook



Chinook Query (1)

What are all the genres whose longest track is longer than 10 minutes (600,000 milliseconds)? Sort them alphabetically.

	gName
1	Alternative
2	Comedy
3	Drama
4	Jazz
5	Metal
6	Pop
7	Rock
8	Sci Fi & Fantasy
9	Science Fiction
10	TV Shows



SQL (1)

```
SELECT
    g.Name AS gName
FROM
    track t INNER JOIN genre g ON t.GenreId=g.GenreId
GROUP BY
    g.GenreId
HAVING
    MAX(t.milliseconds) > 600000
ORDER BY
    gName ASC
```



Chinook Query (2)

List all employees and how many invoices were associated with customers they supported. Sort by number of invoices (biggest to smallest), then by last name & then first name (both alphabetical).

	firstName	lastName	numInvoices
1	Jane	Peacock	146
2	Margaret	Park	140
3	Steve	Johnson	126
4	Andrew	Adams	0
5	Laura	Callahan	0
6	Nancy	Edwards	0
7	Robert	King	0
8	Michael	Mitchell	0



SQL (2)

SELECT

**e.FirstName AS firstName,
e.LastName AS lastName,
COUNT(i.InvoiceId) AS numInvoices**

FROM

**Employee e LEFT JOIN
(Customer c INNER JOIN Invoice i ON c.CustomerId=i.CustomerId)
ON e.EmployeeId=c.SupportRepId**

GROUP BY

e.EmployeeId

ORDER BY

**numInvoices DESC,
lastName ASC,
firstName ASC**



What DB Does This Produce?

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



Query this DB (1)

uno		dos	
<u>a</u>	b	<u>x</u>	<u>y</u>
1	foo	2	1
2	bar	3	1
3	baz	7	4
4	qux	8	3

Predict the outcome of the following query (rows/cols?):

```
SELECT *, (dos.x + dos.y) AS dosTotal
FROM uno INNER JOIN dos ON uno.a=dos.y
ORDER BY uno.b, dos.x DESC
```



Result (1)

a	b	x	y	dosTotal
3	baz	8	3	11
1	foo	3	1	4
1	foo	2	1	3
4	qux	7	4	11



Query this DB (2)

uno		dos	
<u>a</u>	b	<u>x</u>	<u>y</u>
1	foo	2	1
2	bar	3	1
3	baz	7	4
4	qux	8	3

Predict the outcome of the following query (rows/cols?):

```
SELECT b AS stuff, COUNT(*) AS things
FROM uno INNER JOIN dos ON uno.a=dos.y
GROUP BY b
ORDER BY things DESC, stuff ASC
```



Result (2)

stuff	things
foo	2
baz	1
qux	1



Query this DB (3)

uno		dos	
<u>a</u>	b	<u>x</u>	<u>y</u>
1	foo	2	1
2	bar	3	1
3	baz	7	4
4	qux	8	3

Predict the outcome of the following query (rows/cols?):

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



Result (3)

var	avgLine
qux	7.0
foo	2.5
baz	8.0
bar	<i>NULL</i>



Good Luck on Exam 1 :)

