



Introduction  
to Database  
Systems:  
CS305  
A Focus On  
Queries

Oliver  
Bonham-  
Carter  
Hang Zhao

Basic Query  
Structures  
Clauses  
WHERE

CampusDB  
Abbreviations  
Aggregate  
Functions

Removing  
Tables or Data  
Changing Table  
Contents

Your Turn!

# Introduction to Database Systems: CS305

## A Focus On Queries

Oliver Bonham-Carter  
Hang Zhao

26 September 2023

# Let's Spend a Moment to Recap

Introduction  
to Database  
Systems:  
CS305  
A Focus On  
Queries

Oliver  
Bonham-  
Carter  
Hang Zhao

Basic Query  
Structures

Clauses  
WHERE

CampusDB

Abbreviations  
Aggregate  
Functions

Removing  
Tables or Data

Changing Table  
Contents

Your Turn!



NEW FROM KODAK

Automatic Cavalcade leaves you free to enjoy the show!

**CHANGES SLIDES BY ITSELF!**

New Kodak Cavalcade Projector, only \$124.50

You can own a new, lower cost model of the famous \$159.50 Kodak Cavalcade Projector. It has the same automatic action—you turn it on, it does the rest. But this one's only \$124.50.

You put on big, bright shows—up to 40 slides in a row—with no work at all! Your Cavalcade changes slides quietly, dependably, automatically—at the time intervals you choose. You reverse, repeat, edit at will.

Your pictures stay sharp on the screen—no refocusing—because every slide is preconditioned by warm air. Your slides are safe . . . each in its own steel guard for smooth, jam-free showings.

And each slide is projected 500-watts-bright through the brilliant f/3.5 lens.

See the new Kodak Cavalcade Projector, Model 520, demonstrated at your Kodak dealer's. At only \$124.50, it's a superb value. As little as \$12.50 down at many dealers. Model 510 with remote-control cord, and f/2.8 lens for extra brilliance, \$159.50.

Prices are list, include Federal Tax, and are subject to change without notice.

*See Kodak's "The Ed Sullivan Show" and "The Adventures of Ozzie and Harriet"*

EASTMAN KODAK COMPANY, Rochester 4, N.Y.

Kodak

Navigation icons: back, forward, search, etc.

# How to Connect Information?

No free-standing tables allowed!

Introduction  
 to Database  
 Systems:  
 CS305  
 A Focus On  
 Queries

Oliver  
 Bonham-  
 Carter  
 Hang Zhao

Basic Query  
 Structures

Clauses  
 WHERE

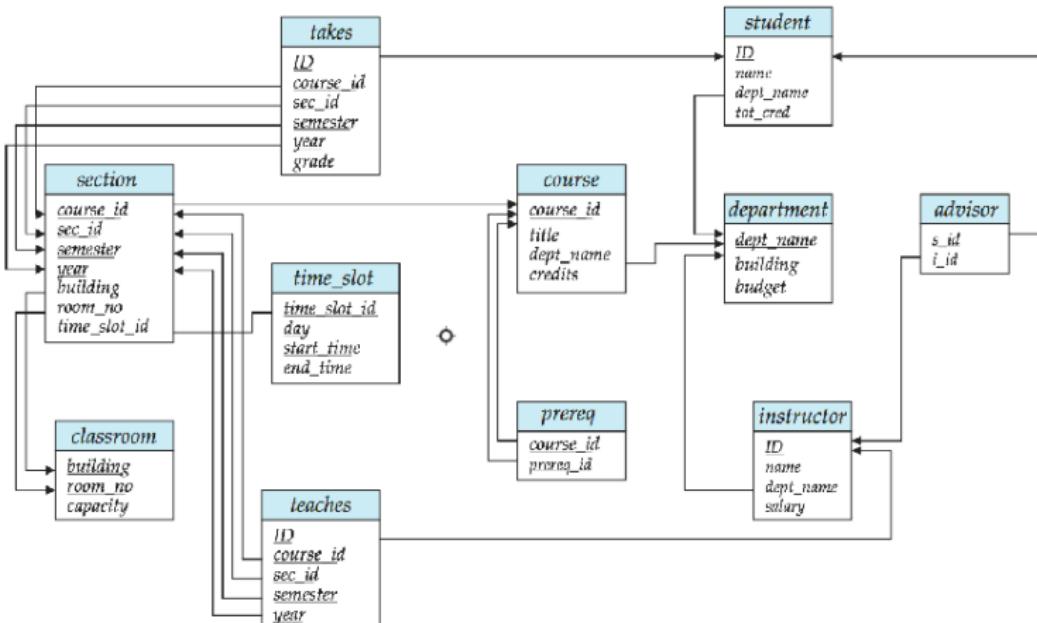
CampusDB

Abbreviations  
 Aggregate  
 Functions

Removing  
 Tables or Data

Changing Table  
 Contents

Your Turn!



# The Basic Query Structure

Introduction  
to Database  
Systems:  
CS305  
A Focus On  
Queries

Oliver  
Bonham-  
Carter  
Hang Zhao

Basic Query  
Structures

Clauses  
WHERE

CampusDB

Abbreviations  
Aggregate  
Functions

Removing  
Tables or Data

Changing Table  
Contents

Your Turn!

The SQL data-manipulation language (DML) provides the ability to query information, and insert, delete and update tuples

A typical SQL pseudo code query has the form:

```
SELECT A1, A2, ..., An  
FROM r1, r2, ..., rm  
WHERE P;
```

- $A_i$  represents an attribute
- $R_i$  represents a relation
- $P$  is a predicate
- The result of an SQL query is a relation

# The **SELECT** Clause

Introduction  
to Database  
Systems:  
CS305  
A Focus On  
Queries

Oliver  
Bonham-Carter  
Hang Zhao

Basic Query  
Structures  
Clauses  
WHERE  
CampusDB  
Abbreviations  
Aggregate  
Functions  
Removing  
Tables or Data  
Changing Table  
Contents

Your Turn!

The **SELECT** clause filters out particular data from a table.

- SQL allows duplicates in relations as well as in query results.
- The **SELECT** statement has many optional clauses:
  - WHERE specifies which rows to retrieve.
  - GROUP BY groups rows sharing a property so that an aggregate function can be applied to each group.
  - HAVING selects among the groups defined by the GROUP BY clause.
  - ORDER BY specifies an order in which to return the rows.
  - AS provides an alias which can be used to temporarily rename tables or columns..

# Given table 'T'

Introduction  
to Database  
Systems:  
CS305  
A Focus On  
Queries

Oliver  
Bonham-  
Carter  
Hang Zhao

Basic Query  
Structures  
Clauses  
WHERE

CampusDB  
Abbreviations  
Aggregate  
Functions

Removing  
Tables or Data  
Changing Table  
Contents

Your Turn!

Table "T"	Query	Result												
<table border="1"> <thead> <tr> <th>C1</th><th>C2</th></tr> </thead> <tbody> <tr> <td>1</td><td>a</td></tr> <tr> <td>2</td><td>b</td></tr> </tbody> </table>	C1	C2	1	a	2	b	<code>SELECT * FROM T;</code>	<table border="1"> <thead> <tr> <th>C1</th><th>C2</th></tr> </thead> <tbody> <tr> <td>1</td><td>a</td></tr> <tr> <td>2</td><td>b</td></tr> </tbody> </table>	C1	C2	1	a	2	b
C1	C2													
1	a													
2	b													
C1	C2													
1	a													
2	b													
<table border="1"> <thead> <tr> <th>C1</th><th>C2</th></tr> </thead> <tbody> <tr> <td>1</td><td>a</td></tr> <tr> <td>2</td><td>b</td></tr> </tbody> </table>	C1	C2	1	a	2	b	<code>SELECT C1 FROM T;</code>	<table border="1"> <thead> <tr> <th>C1</th></tr> </thead> <tbody> <tr> <td>1</td></tr> <tr> <td>2</td></tr> </tbody> </table>	C1	1	2			
C1	C2													
1	a													
2	b													
C1														
1														
2														
<table border="1"> <thead> <tr> <th>C1</th><th>C2</th></tr> </thead> <tbody> <tr> <td>1</td><td>a</td></tr> <tr> <td>2</td><td>b</td></tr> </tbody> </table>	C1	C2	1	a	2	b	<code>SELECT * FROM T WHERE C1 = 1;</code>	<table border="1"> <thead> <tr> <th>C1</th><th>C2</th></tr> </thead> <tbody> <tr> <td>1</td><td>a</td></tr> </tbody> </table>	C1	C2	1	a		
C1	C2													
1	a													
2	b													
C1	C2													
1	a													
<table border="1"> <thead> <tr> <th>C1</th><th>C2</th></tr> </thead> <tbody> <tr> <td>1</td><td>a</td></tr> <tr> <td>2</td><td>b</td></tr> </tbody> </table>	C1	C2	1	a	2	b	<code>SELECT * FROM T ORDER BY C1 DESC;</code>	<table border="1"> <thead> <tr> <th>C1</th><th>C2</th></tr> </thead> <tbody> <tr> <td>2</td><td>b</td></tr> <tr> <td>1</td><td>a</td></tr> </tbody> </table>	C1	C2	2	b	1	a
C1	C2													
1	a													
2	b													
C1	C2													
2	b													
1	a													

# The TeaDB

Introduction  
to Database  
Systems:

CS305

A Focus On  
Queries

Oliver  
Bonham-  
Carter  
Hang Zhao

Basic Query  
Structures

Clauses

WHERE

CampusDB

Abbreviations

Aggregate  
Functions

Removing  
Tables or Data

Changing Table  
Contents

Your Turn!

Build file: sandbox/teaDB/teaDB\_Build.txt

```
cat builder_teaDB.txt | sqlite3 teaDB.sqlite3
```

(or just copy in the text into SQLite3!)





# The **SELECT** Clause

## TeaDB

Introduction  
to Database  
Systems:  
CS305  
A Focus On  
Queries

Oliver  
Bonham-  
Carter  
Hang Zhao

Basic Query  
Structures

Clauses

WHERE

CampusDB

Abbreviations  
Aggregate  
Functions

Removing  
Tables or Data

Changing Table  
Contents

Your Turn!

- Find everything in the Department table.

- `SELECT * FROM Department;`

- Find all entries for *dept*'s of the Department table

- `SELECT dept from Department;`

- Count entries of *dept*'s in Department table,

- `SELECT COUNT(dept) FROM department;`



# The SELECT Clause

## TeaDB

Introduction  
to Database  
Systems:  
CS305

A Focus On  
Queries

Oliver  
Bonham-  
Carter  
Hang Zhao

Basic Query  
Structures  
Clauses  
WHERE

CampusDB  
Abbreviations  
Aggregate  
Functions

Removing  
Tables or Data

Changing Table  
Contents

Your Turn!

- Find all unique entries for *depts* in Department table,
- SELECT DISTINCT(dept) FROM department;
- Count unique entries of *depts* in Department table,
- SELECT COUNT(DISTINCT(dept)) FROM Department;  
/\*count unique occurrences\*/
- Return an exhaustive set of sandwiches that are being ordered.
- SELECT DISTINCT(sandwich) FROM Tea;
- What query to use to count these types of sandwiches?



# The WHERE clause

## TeaDB

Introduction  
to Database  
Systems:  
CS305  
A Focus On  
Queries

Oliver  
Bonham-  
Carter  
Hang Zhao

Basic Query  
Structures  
Clauses  
WHERE

CampusDB  
Abbreviations  
Aggregate  
Functions

Removing  
Tables or Data  
Changing Table  
Contents

Your Turn!

### The WHERE clause: **conditions** that the result must satisfy

- Corresponds to the selection predicate of the relational algebra
- Comparison results can be combined using the logical connectives and, or, and not
- Comparisons can be applied to results of arithmetic expressions



# The WHERE clause

## TeaDB

Introduction  
to Database  
Systems:  
CS305  
A Focus On  
Queries

Oliver  
Bonham-  
Carter  
Hang Zhao

Basic Query  
Structures

Clauses  
WHERE

CampusDB

Abbreviations  
Aggregate  
Functions

Removing  
Tables or Data

Changing Table  
Contents

Your Turn!

- Find out who is ordering a sandwich less than \$15 (from the new cost column)

● `SELECT * FROM tea WHERE cost < 15;`

- Find out what kinds of *sandwiches* are going to each *dept*

● `SELECT department.dept, tea.sandwich FROM department, tea  
WHERE department.id == tea.id;`



# The WHERE clause

## TeaDB

Introduction  
to Database  
Systems:  
CS305  
A Focus On  
Queries

Oliver  
Bonham-  
Carter  
Hang Zhao

Basic Query  
Structures  
Clauses  
WHERE

CampusDB  
Abbreviations  
Aggregate  
Functions

Removing  
Tables or Data  
Changing Table  
Contents

Your Turn!

- Find department, Session material, sandwich type for orders of sandwiches less than \$15.

- SELECT

```
Department.id, Session.session,  
tea.sandwich, tea.cost  
FROM  
    Tea, Department, Session  
WHERE  
    cost < 15  
AND  
    Department.id == Session.id  
AND  
    Department.id == Tea.id;
```



# The WHERE clause

## TeaDB

Introduction  
to Database  
Systems:  
CS305  
A Focus On  
Queries

Oliver  
Bonham-  
Carter  
Hang Zhao

Basic Query  
Structures

Clauses  
WHERE

CampusDB

Abbreviations  
Aggregate  
Functions

Removing  
Tables or Data

Changing Table  
Contents

Your Turn!

- Find out which professors are presenting posters
- `SELECT * FROM session WHERE material == "poster"; /* show all*/`
- `SELECT ID, material FROM session WHERE material == "poster"; /*which professor is doing what?*/`
- Find how who is presenting a poster, having what kind of sandwich which costs over \$10
- `SELECT session.ID, session.material, tea.sandwich, tea.cost FROM session, tea WHERE session.material == "poster" AND tea.cost > 10 AND session.id == tea.id;`



# Moving On

## And now this!

Introduction  
to Database  
Systems:  
CS305  
A Focus On  
Queries

Oliver  
Bonham-  
Carter  
Hang Zhao

Basic Query  
Structures  
Clauses  
WHERE  
  
CampusDB  
Abbreviations  
Aggregate  
Functions  
  
Removing  
Tables or Data  
Changing Table  
Contents  
  
Your Turn!



# New Database Tables!

## CampusDB

Introduction  
to Database

Systems:  
CS305

A Focus On  
Queries

Oliver  
Bonham-  
Carter  
Hang Zhao

Basic Query  
Structures

Clauses  
WHERE

CampusDB

Abbreviations  
Aggregate  
Functions

Removing  
Tables or Data

Changing Table  
Contents

Your Turn!

Build file: `sandbox/campusDB/builder_campusDB`

```
cat campusDB_build.txt | sqlite3 CampusDB.sqlite3
```

(or just copy in the text into SQLite3!)





# Abbreviations in Queries

## CampusDB

Introduction  
to Database  
Systems:  
CS305  
A Focus On  
Queries

Oliver  
Bonham-  
Carter  
Hang Zhao

Basic Query  
Structures  
Clauses  
WHERE  
CampusDB  
Abbreviations  
Aggregate  
Functions  
Removing  
Tables or Data  
Changing Table  
Contents  
Your Turn!

Find which students are working with what instructors.

- `SELECT Instructor.ID, Instructor.name, Instructor.studentId, Student.name, Student.Id FROM Instructor, Student WHERE Instructor.studentId == Student.ID;`

Shorter way to write query by using abbreviations

- `SELECT i.ID, i.name, i.studentId, s.name, s.Id FROM Instructor i, Student s WHERE i.studentId == s.ID;`

- The “**Instructor**” table name can be replaced with an **i**
- The “Student” table name can be replaced by an “s”



# Aggregate Functions

## CampusDB

Introduction  
to Database  
Systems:  
CS305  
A Focus On  
Queries

Oliver  
Bonham-  
Carter  
Hang Zhao

Basic Query  
Structures

Clauses  
WHERE

CampusDB  
Abbreviations  
Aggregate  
Functions

Removing  
Tables or Data  
Changing Table  
Contents

Your Turn!

These functions operate on the multiset of values of a column of a relation, and return a value

- **avg**: average value
- **min**: minimum value
- **max**: maximum value
- **sum**: sum of values
- **count**: number of values



# Mathematical Functions

## CampusDB

Introduction  
to Database  
Systems:  
CS305

A Focus On  
Queries

Oliver  
Bonham-  
Carter  
Hang Zhao

Basic Query  
Structures

Clauses  
WHERE

CampusDB

Abbreviations  
Aggregate  
Functions

Removing  
Tables or Data

Changing Table  
Contents

Your Turn!

To find all instructors in Comp. Sci. dept with salary > 80000

- SELECT  
    name FROM instructor  
    WHERE  
        deptName = "CompSci"  
    AND  
        salary > 80000;

Using built-in functions

- SELECT AVG (salary) FROM instructor  
    WHERE deptName = "CompSci";
- SELECT MIN (salary) FROM instructor  
    WHERE deptName = "CompSci";
- SELECT MAX (salary) FROM instructor  
    WHERE deptName = "CompSci";



# Mathematical Functions

## CampusDB

Introduction  
to Database  
Systems:  
CS305  
A Focus On  
Queries

Oliver  
Bonham-  
Carter  
Hang Zhao

Basic Query  
Structures  
Clauses  
WHERE  
CampusDB  
Abbreviations  
Aggregate  
Functions  
Removing  
Tables or Data  
Changing Table  
Contents  
Your Turn!

To find all instructors in Comp. Sci. dept with salary > 80000

- ```
SELECT name FROM instructor
      WHERE deptName = "CompSci" AND salary > 80000;
```

Using functions

- ```
SELECT SUM (salary) FROM instructor
      WHERE deptName = "CompSci";
```
- ```
SELECT COUNT (salary) FROM instructor
      WHERE deptName = "CompSci";
```

# Attention to the WHERE clause

## CampusDB

Introduction  
to Database  
Systems:  
CS305  
A Focus On  
Queries

Oliver  
Bonham-  
Carter  
Hang Zhao

Basic Query  
Structures  
Clauses  
WHERE  
CampusDB  
Abbreviations  
Aggregate  
Functions  
Removing  
Tables or Data  
Changing Table  
Contents  
Your Turn!

- Find the ID, name and total credit students who are taking a course where the total credit is 3 or 4 hours.

### Why will “AND” NOT work

- `SELECT ID, name, totCred FROM student WHERE totCred == "3" OR totCred == "4";`

### Watch out for cross products that give no usable information!!

- `SELECT s.name, i.name from student s, instructor i WHERE s.deptName == i.deptName and s.deptName == "CompSci";`
- Common Solution – Use two queries instead
  - `SELECT s.name from student s WHERE s.deptName == "CompSci";`
  - `SELECT i.name from instructor i WHERE i.deptName == "CompSci";`



# Using Count and Count(Distinct(...))

## CampusDB

Introduction  
to Database  
Systems:  
CS305  
A Focus On  
Queries

Oliver  
Bonham-  
Carter  
Hang Zhao

Basic Query  
Structures  
Clauses  
WHERE  
CampusDB  
Abbreviations  
Aggregate  
Functions

Removing  
Tables or Data  
Changing Table  
Contents

Your Turn!

Find the number of tuples in the course relation

- `SELECT COUNT(credits) FROM course;`
- `SELECT COUNT(distinct(credits)) FROM course;`
- `SELECT COUNT (*) FROM course;`
- **`SELECT COUNT(distinct(*)) FROM course;`**
- Question: Why will the above *distinct* line **not** work?



# Removing Tables or Data

CampusDB: Adding data to Student table

Introduction  
to Database  
Systems:  
CS305  
A Focus On  
Queries

Oliver  
Bonham-  
Carter  
Hang Zhao

Basic Query  
Structures  
Clauses  
WHERE

CampusDB  
Abbreviations  
Aggregate  
Functions

Removing  
Tables or Data

Changing Table  
Contents

Your Turn!

- **DROP TABLE IF EXISTS student**
  - Deletes the table *student* and its contents if present
- **DROP TABLE student**
  - Deletes the table *student* and its contents, report error if table not present
- **DELETE FROM student**
  - Deletes all contents from table *student*, but retains table

Play with your database!

Remember, you can use your builder file to re-create the database if it becomes corrupt or unstable.

# Changing Table Contents

Introduction  
to Database  
Systems:  
CS305  
A Focus On  
Queries

Oliver  
Bonham-  
Carter  
Hang Zhao

Basic Query  
Structures  
Clauses  
WHERE  
CampusDB  
Abbreviations  
Aggregate  
Functions  
Removing  
Tables or Data  
Changing Table  
Contents  
Your Turn!

## ● ALTER TABLE

- Alter TABLE  $r$  ADD  $AD$
- where  $A$  is the name of the attribute to be added to relation  $r$  and  $D$  is the domain of  $A$ .
- All tuples in the relation are assigned null as the value for the new attribute.
- EX: ALTER TABLE Department ADD Email varchar;

## ● Change name of table:

- ALTER TABLE Department RENAME TO newDept;

# Changing Table Contents

Introduction  
to Database  
Systems:  
CS305  
A Focus On  
Queries

Oliver  
Bonham-  
Carter  
Hang Zhao

Basic Query  
Structures

Clauses  
WHERE

CampusDB

Abbreviations  
Aggregate  
Functions

Removing  
Tables or Data

Changing Table  
Contents

Your Turn!

- **Add** a column to a table

- `ALTER TABLE course ADD COLUMN courseTag VARCHAR;`
- Check your additional column:
  - `.schema course`

- **Dropping** a column of a table

- `ALTER TABLE course DROP COLUMN courseTag;`
- Check your additional column:
  - `.schema course`



# Complex Queries

Instructor names, IDs and their Students?

Introduction  
to Database  
Systems:  
CS305

A Focus On  
Queries

Oliver  
Bonham-  
Carter  
Hang Zhao

Basic Query  
Structures

Clauses  
WHERE

CampusDB

Abbreviations  
Aggregate  
Functions

Removing  
Tables or Data

Changing Table  
Contents

Your Turn!

SELECT

```
instructor.name, instructor.id,  
instructor.studentID, student.ID,  
student.name
```

FROM

```
instructor, student
```

WHERE

```
student.id == instructor.studentID;
```

## Output

```
Miller|10101|S1|S1|Michaels  
Johnson|10102|S1|S1|Michaels  
Charleson|10103|S2|S2|Peterson  
Thompson|10104|S2|S2|Peterson  
Mauler|10105|S3|S3|Mullen  
...  
Farber|10112|S5|S5|Beuller
```



# Try these with campusDB.sqlite3!

Work by yourself or together with others

Introduction  
to Database  
Systems:  
CS305  
A Focus On  
Queries

Oliver  
Bonham-  
Carter  
Hang Zhao

Basic Query  
Structures

Clauses  
WHERE

CampusDB  
Abbreviations  
Aggregate  
Functions

Removing  
Tables or Data  
Changing Table  
Contents

Your Turn!

- What is the average salary of computer science teachers?
- What is the average salary of computer science teachers who make less than \$98000?

**THINK**



# Try these with campusDB.sqlite3!

Work by yourself or together with others

Introduction  
to Database  
Systems:  
CS305  
A Focus On  
Queries

Oliver  
Bonham-  
Carter  
Hang Zhao

Basic Query  
Structures  
Clauses  
WHERE

CampusDB  
Abbreviations  
Aggregate  
Functions

Removing  
Tables or Data  
Changing Table  
Contents

Your Turn!

- What are the average salaries of instructors who worked during the Spring?
- What are the average salaries of instructors who worked during the Fall?

**THINK**



# Try these with campusDB.sqlite3!

Work by yourself or together with others

Introduction  
to Database  
Systems:  
CS305  
A Focus On  
Queries

Oliver  
Bonham-  
Carter  
Hang Zhao

Basic Query  
Structures  
Clauses  
WHERE

CampusDB  
Abbreviations  
Aggregate  
Functions  
Removing  
Tables or Data  
Changing Table  
Contents  
Your Turn!

- What are the Instructor names and their IDs who taught which Students (show names and IDs)?
- What are the Instructor IDs their Student's IDs in cases where the instructors and students are **NOT** in the same department?

THINK



# Try these with campusDB.sqlite3!

Work by yourself or together with others

Introduction  
to Database  
Systems:

CS305

A Focus On  
Queries

Oliver  
Bonham-  
Carter  
Hang Zhao

Basic Query  
Structures

Clauses

WHERE

CampusDB

Abbreviations

Aggregate  
Functions

Removing  
Tables or Data

Changing Table  
Contents

Your Turn!

- What are the Instructor names and their IDs who taught which Students (show names and IDs) for classes taught in the year 2010?
- What are the Instructor names and their IDs who taught which Students (show names and IDs) for classes taught in the year 2010. In which semester were they teaching?
- Come-up with your own complex question and query solution.

THINK