

SQL

SQL

- SQL is the standard language for interacting with RDBMSes
 - supported by DBMSes from many different vendors (though there are incompatibilities due to incomplete implementations and non-standard extensions)
- includes multiple sublanguages
 - data query language (DQL) – for retrieving data
 - data manipulation language (DML) – insert, update, delete data
 - data definition language (DDL) – define tables and other DB objects
 - data control language (DCL) – manage user permissions and authorization

Relational Databases

- a relational database consists of tables
- each row in a table groups together data values about a particular thing
 - duplicate rows are not allowed (must differ by at least one value)
- primary key** – column(s) which together uniquely identify rows
 - two rows cannot have the same values for all the fields of the PK – PK uniquely identifies each row
- foreign key** – column values reference rows in another table
- notation for relational schemas
 - underline denotes the PK
 - foreign keys are shown with arrows or listed separately

EMPLOYEE								
Empno	Minit	Lname	Sex	Bdate	Address	Sex	Salary	Super_ssn
John	B	Smith	M	1940-01-09	731 Fonders, Houston, TX	M	30000	333445555
Franklin	T	Wong	M	1950-12-08	638 Vines, Houston, TX	M	40000	888885555
Alice	J	Zapp	F	1980-01-13	3021 Castle, Spring, TX	F	20000	987654321
Jennifer	S	Walton	F	1941-09-20	291 Burg, Dallas, TX	F	43000	888885555
Ramesh	K	Nagaya	M	1952-08-15	876 Pine Oak, Houston, TX	M	38000	333445555
John	A	English	M	1972-01-31	1831 Rice, Houston, TX	F	20000	333445555
Alfred	V	Jahner	M	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321
James	E	Borg	M	1937-11-10	450 Stone, Houston, TX	M	50000	NULL

DEPARTMENT			
Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1988-01-01
Headquarters	1	888885555	1981-06-19

DEPT_LOCATIONS			
Dnumber	Location		
1	Houston		
4	Stafford		
5	Dallas		
5	Stafford		
5	Houston		

WORKS_ON		
Emp	Dno	Hours
123456789	1	35.5
123456789	2	7.5
888884444	5	40.0
456789012	1	20.0
456789012	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0

PROJECT				
Pname	Pnumber	Plocation	Pstart	Pdur
ProjectA	1	Dallas	5	
ProjectB	2	Stafford	5	
ProjectC	3	Houston	5	
ProjectD	10	Stafford	4	
ProjectE	20	Houston	1	
ProjectF	30	Stafford	4	

DEPENDENT				
Esan	Dependent_name	Sex	Bdate	Relationship
333445555	Alice	F	1988-04-05	Daughter
333445555	Therese	M	1989-10-25	Son
987654321	Jay	F	1958-06-03	Spouse
987654321	Alfred	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1987-05-05	Spouse

EMPLOYEE (Fname, Minit, Lname, Ssn, Bdate, Address, Sex, Salary, Super_ssn, Dno)
 DEPARTMENT (Dname, Dnumber, Mgr_ssn, Mgr_start_date)
 WORKS_ON (Esan, Pno, Hours)
 ...

Bags vs Sets

- in the relational model, relations are **sets**
 - no duplicate tuples
 - order of tuples doesn't matter
- in practice, tables are implemented as **bags**
 - order of rows doesn't matter
 - duplicates are not automatically removed from query results*

One possible database state for the COMPANY relational database schema. **Figure 5.6**

EMPLOYEE										
fname	lname	ssn	bdate	address	sex	salary	superssn	deptnum		
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5	
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5	
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4	
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4	
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5	
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5	
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4	
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1	

DEPARTMENT			
deptname	deptnum	mgrssn	mgrstartdate
Research	5	333445555	1955-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS	
deptnum	deptloc
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

WORKS_ON		
empssn	projnum	hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	20.0

PROJECT					
projname	projnum	projloc	deptnum		
ProductX	1	Bellaire	5		
ProductY	2	Sugarland	5		
ProductZ	3	Houston	5		
Computerization	10	Stafford	4		
Reorganization	20	Houston	1		
Newbenefits	30	Stafford	4		

DEPENDENT					
empssn	fname	sex	bdate	relationship	
333445555	Alice	F	1986-04-05	Daughter	
333445555	Theodore	M	1983-10-25	Son	
333445555	Joy	F	1958-05-03	Spouse	
987654321	Abner	M	1942-02-28	Spouse	
123456789	Michael	M	1988-01-04	Son	
123456789	Alice	F	1988-12-30	Daughter	
123456789	Elizabeth	F	1967-05-05	Spouse	

CPSC 343: Database Theory • Fall 2026 5

Basic Queries

```
SELECT a1,a2,...,an
FROM R
WHERE C
```

- (only) table and column names are case-sensitive
- resulting table has
 - the rows from R for which C is true
 - the columns a1,a2,...,an

Basic Queries

```
SELECT fname, lname, salary
FROM EMPLOYEE
WHERE deptnum = 5
```

names and salaries of employees in department 5

```
SELECT fname, lname, salary
FROM EMPLOYEE
WHERE salary > 60000
```

names and salaries of employees who earn more than 60,000

```
SELECT fname
FROM DEPENDENT
WHERE sex='F'
```

first names of female dependents

```
SELECT fname, lname, sex, salary
FROM EMPLOYEE
WHERE sex = 'F' AND salary >= 30000
```

name, sex, and salary of female employees earning at least 30,000

```
SELECT projname, projloc, deptnum
FROM PROJECT
WHERE projloc = 'Houston' OR deptnum = 5
```

name, location, and department for projects located in Houston or in department 5

```
SELECT deptname, deptnum
FROM DEPARTMENT
WHERE deptnum <= 5
```

department name and number for departments whose number is at most 5

Basic Queries

```
SELECT empssn, fname, relationship, bdate
FROM DEPENDENT
WHERE (relationship = 'Son' OR relationship = 'Daughter') AND
      bdate > '1970-01-01'
```

first name, birthdate, relationship, and employee for employee children (son or daughter) born after Jan 1 1970

```
SELECT empssn, projnum, hours
FROM WORKS_ON
WHERE hours >= 20 AND hours <= 40
```

employee (ssn), project (number), and hours worked for which the hours worked is between 20 and 40 (inclusive)

```
SELECT deptnum, deptloc
FROM DEPT_LOCATIONS
WHERE deptloc = 'Chicago' OR deptloc = 'Dallas'
```

department number and location for departments with locations in Chicago or Dallas

```
SELECT projname, projnum, deptnum
FROM PROJECT
WHERE projnum > 60 AND projnum < 90
```

projects (name, number, and department number) for project numbers between 60 and 90 (exclusive)

Basic Queries

SELECT

- can contain * (all columns), arithmetic expressions
- can rename individual columns of the result using AS

WHERE

- can contain arithmetic expressions, <, <=, >, >=, =, <>, AND, OR, NOT, IS NULL, IS NOT NULL, LIKE, NOT LIKE
 - wildcards % (0 or more), _ (single character) allowed in LIKE/NOT LIKE patterns – used for pattern-matching within column values

Basic Queries

```
SELECT fname, lname, salary, salary * 1.10 AS newsalary
FROM EMPLOYEE
WHERE deptnum = 6
```

name, salary, and salary including a 10% bonus for employees in department 6

```
SELECT *
FROM EMPLOYEE
WHERE salary > 50000
```

employees who earn more than 50,000

```
SELECT fname, lname
FROM EMPLOYEE
WHERE lname LIKE 'K%'
```

names of employees whose last name starts with K

```
SELECT fname, lname
FROM EMPLOYEE
WHERE lname NOT LIKE 'B%'
```

names of employees whose last name doesn't start with B

```
SELECT fname, lname, address
FROM EMPLOYEE
WHERE address LIKE '%St%'
```

names and addresses of employees whose address includes St

```
SELECT empssn, projnum, hours, hours * 52 AS yearlyhours
FROM WORKS_ON
WHERE hours > 30
```

employee (ssn), project (number), and weekly and projected yearly (weekly*52) hours where the weekly hours is over 30

Basic Queries

```
SELECT fname, lname, salary
FROM EMPLOYEE
WHERE salary / 12 < 3000
```

name and salary for employees whose monthly salary is below 3,000

```
SELECT fname, lname, salary, salary / 12 AS monthllysalary
FROM EMPLOYEE
WHERE salary >= 40000
```

name, annual and monthly salaries for employees earning at least 40,000 annually

```
SELECT fname, lname, salary, salary - 50000 AS amountoverthreshold
FROM EMPLOYEE
WHERE salary > 50000
ORDER BY amountoverthreshold DESC
```

for employees whose salary exceeds a 50,000 threshold, their name, salary, and the amount over threshold, from highest to lowest

```
SELECT fname, lname
FROM EMPLOYEE
WHERE fname LIKE 'J____'
```

names of employees whose first name is a four-letter name starting with J

```
SELECT fname, lname, bdate
FROM EMPLOYEE
WHERE bdate LIKE '197_09-__'
```

name and birthdate of employees born in September during the 1970s

```
SELECT fname, lname, bdate
FROM EMPLOYEE
WHERE bdate LIKE '____-__-31'
```

name and birthdate of employees born on the 31st of a month

Basic Queries

Notes on data types.

- ' ' (single quotes) denote string values
- DATE – 'yyyy-mm-dd'
- TIME – 'hh:mm:ss' or 'hh:mm:ss+h:mm' or 'hh:mm:ss-h:mm' (relative to GMT)
- TIMESTAMP – 'yyyy-mm-dd hh:mm:ss'

NULL values.

- any operation applied to NULL yields NULL
- any comparison with NULL yields UNKNOWN

Three-valued logic.

- works out as if TRUE = 1, FALSE = 0, UNKNOWN = ½ and AND = min, OR = max, NOT = 1-value

Basic Queries

```
SELECT fname, lname, ssn
FROM EMPLOYEE
WHERE superssn IS NULL
```

name and SSN of employees with no supervisor

```
SELECT fname, lname, superssn
FROM EMPLOYEE
WHERE superssn IS NOT NULL
```

name and supervisor's SSN for employees with a supervisor

```
SELECT deptname, deptnum
FROM DEPARTMENT
WHERE mgrssn IS NULL
```

department name and number for departments with no manager

Basic Queries

```
SELECT a1,a2,...,an
```

```
FROM R
```

```
WHERE C
```

```
ORDER BY b1,b2,...,bm
```

- rows are sorted according to the specified columns just before the SELECT is applied (so b1,b2,...,bm can involve columns other than a1,a2,...,an)
- default is ascending order; use DESC for descending order

```
SELECT DISTINCT a1,a2,...,an
```

```
FROM R
```

```
WHERE C
```

- as the last step, remove duplicate rows (duplicates identified based only on columns a1,a2,...,an)

Basic Queries

```
SELECT fname, lname, salary
FROM EMPLOYEE
WHERE salary > 30000
ORDER BY salary
```

name and salary of employees earning more than 30,000, in increasing order by salary

```
SELECT fname, lname
FROM EMPLOYEE
WHERE deptnum = 8
ORDER BY lname ASC
```

names of employees in department 8, in increasing order by last name

```
SELECT fname, lname, salary
FROM EMPLOYEE
WHERE deptnum = 7
ORDER BY salary DESC
```

name and salary of employees in department 7, in decreasing order by salary

```
SELECT fname, lname, deptnum, salary
FROM EMPLOYEE
WHERE salary > 25000
ORDER BY deptnum ASC, salary DESC
```

name, department number, and salary of employees earning more than 25,000, in increasing order by department and decreasing order of salary within departments

Basic Queries

```
SELECT DISTINCT sex
FROM EMPLOYEE
WHERE deptnum = 8
```

which sexes are represented by the employees in department 8 (unique values)

```
SELECT DISTINCT deptnum
FROM PROJECT
ORDER BY deptnum DESC
```

departments (number) which have associated projects, in decreasing order of department number (unique values)

```
SELECT DISTINCT fname
FROM DEPENDENT
WHERE relationship = 'Spouse'
```

first names of spouses (unique values)

```
SELECT DISTINCT fname, sex
FROM DEPENDENT
WHERE relationship = 'Spouse'
```

first name and sex of spouses (unique combinations)

Syntax Notation

- **blue** (and not italics) – type literally that
- *italics* – replace with the appropriate specifics of that kind of thing
- `|` - list of alternatives, choose one
- `[]` - optional
- `{ }` - provides grouping when there are alternatives for a required thing (don't type the `{ }` themselves)

```
SELECT
[ALL | DISTINCT | DISTINCTROW ]
[HIGH_PRIORITY]
[STRAIGHT_JOIN]
[SQL_SMALL_RESULT] [SQL_BIG_RESULT] [SQL_BUFFER_RESULT]
[SQL_NO_CACHE] [SQL_CALC_FOUND_ROWS]
select_expr [, select_expr] ...
[into_option]
[FROM table_references
[PARTITION partition_list]]
[WHERE where_condition]
[GROUP BY {col_name | expr | position}, ... [WITH ROLLUP]]
[HAVING where_condition]
[WINDOW window_name AS (window_spec)
[, window_name AS (window_spec)] ...]
[ORDER BY {col_name | expr | position}
[ASC | DESC], ... [WITH ROLLUP]]
[LIMIT {[offset,] row_count | row_count OFFSET offset}]
[into_option]
[FOR {UPDATE | SHARE}
[OF tbl_name [, tbl_name] ...]
[NOWAIT | SKIP LOCKED]
| LOCK IN SHARE MODE]
[into_option]

into_option: {
  INTO OUTFILE 'file_name'
  [CHARACTER SET charset_name]
  export_options
| INTO DUMPFILE 'file_name'
| INTO var_name [, var_name] ...
}
```

Basic Queries

- Show the distinct departments that have at least one project located somewhere other than Houston.
- List the distinct sex and relationship combinations found among all dependents, so HR can see what categories of family members are on file.
- Show the unique job locations where companies have projects, excluding any location containing the word "town".
- Find all employees whose last name starts with "K" or ends with "g", sorted alphabetically by last name.
- Find every employee who does not currently have a supervisor assigned, along with their department number, sorted by department.
- List all dependents whose relationship to the employee has never been recorded, along with the employee's SSN.

Basic Queries

- List every department along with its manager's start date, but only for managers who started on or after January 1, 1990.
- List all employees who would earn more than \$70,000 per year after a 15% bonus, showing their name, salary, and bonused salary.
- List employees who make less than \$40,000 a year, showing their name and their approximate biweekly paycheck amount, ordered from lowest to highest salary.
- List the first name, last name, and salary of every employee in the Hardware department (department 7), along with what their salary would be with a 5% raise, sorted from highest current salary to lowest.
- Show every work assignment where an employee would exceed 40 hours on a single project once you add a projected 5-hour increase, along with what that increased total would be.
- Find employees whose address is in Texas (state abbreviation "TX") and whose salary is at least \$30,000, sorted by salary descending and then by last name ascending for any ties.