Database Basics: Relational Databases

These slides have been collected from a variety of sources, including your instructor.

Relational Database

- Definition:
  - Data stored in tables that are associated by shared attributes (keys).
  - Any data element (or entity) can be found in the database through the name of the table, the attribute name, and the value of the primary key.

Relational Database Definitions

- Entity: Object, Concept or event (subject)
- Attribute: a Characteristic of an entity
- Row or Record: the specific characteristics of one entity
- Table: a collection of records
- Database: a collection of tables

The Relational Database model

- Developed by E.F. Codd, C.J. Date (70s)
- Table = Entity = Relation
- Table row = tuple = instance
- Table column = attribute
- Table linkage by values
- Entity-Relationship Model

The Relational Model

- Each attribute has a unique name within an entity
- All entries in the column are examples of it
- Each row is unique
- Ordering of rows and columns is unimportant
- Each position (tuple) is limited to a single entry.

Data Model: What's a model?

- A data model is a representation of reality
- It's used to define the storage and manipulation of a data base.
- Data Models have two components:
  - Structure: the structure of the data stored within
  - Operations: Facilities for manipulation of the data.

CRUD

- Refers to the most common Database Operations:
  - Create
  - Read
  - Update
  - Delete
- Operations occur at all levels: Tables, Records, Columns

Database Tables

- Tables represent entities
- Tables are always named in the singular, such as: Vehicle, Order, Grade, etc.
- Tables in database jargon are “flat files”, dBase or Spreadsheet like..

Attributes

- Characteristics of an entity
- Examples:
  - Vehicle (VIN, color, make, model, mileage)
  - Student (SSN, Fname, Lname, Address)
  - Fishing License (Type, Start_date, End_date)

Database Table Example

Students : Table

	Name	DOB	Address	Phone Number
	Adam Glasnost	1/3/84	11 Price Lane, Spline, 2988	(222) 489-2019
	James Smith	5/9/82	27 Lexington Dr, Ester, 2980	(270) 298-0928
	Clarissa Dubai	12/11/82	123 Dress Dr, Cranton, 2987	(222) 494-2222
	James Smith	5/9/82	5/15 Coventry, Spline, 2988	(278) 509-3981
*				

Record: 2 of 4

Figure 1: A simple – and flawed – table design.

Students II : Table

	StudentID	Lastname	Firstname	DOB	Street address	Suburb	Postcode
	89120	Glasnost	Adam	1/3/84	11 Price Lane	Spline	2988
	90109	Dubai	Clarissa	12/11/82	123 Dress Dr	Cranton	2987
	91082	Smith	James	5/9/82	27 Lexington Dr	Ester	2980
	93007	Smith	James	5/9/82	5/15 Coventry	Spline	2988
▶							

Record: 5 of 5

Figure 2: An improved database table..

Database Views

- A View is an individual's picture of a database. It can be composed of many tables, unbeknownst to the user.
  - It's a simplification of a complex data model
  - It provides a measure of database security
  - Views are useful, primarily for READ-only users and are not always safe for CREATE, UPDATE, and DELETE.

Table Indexing

- An Index is a means of expediting the retrieval of data.
- Indexes are “built” on a column(s).
- Indexes occupy disk space; occasionally a lot.
- Indexes aren’t technically necessary for operation and must be maintained by the database administrator.

Database Table Keys

Definition:
A key of a relation is a subset of attributes with the following attributes:

- Unique identification
- Non-redundancy

Types of Keys

PRIMARY KEY

- Serves as the row level addressing mechanism in the relational database model.
- It can be formed through the combination of several items.

FOREIGN KEY

- A column or set of columns within a table that are required to match those of a primary key of a second table.

These keys are used to form a RELATIONAL JOIN - thereby connecting row to row across the individual tables.

Relational Database Management System (RDBMS)

Table A

Name	Address	Parcel #
John Smith	18 Lawyers Dr.	756554
T. Brown	14 Summers Tr.	887419

Table B

Parcel #	Assessed Value
887419	152,000
446397	100,000

Database Keys

- Primary Key - Indicates uniqueness within records or rows in a table.
- Foreign Key - the primary key from another table, this is the **only** way join relationships can be established.
- There may also be *alternate* or *secondary* keys within a table.

Constructing Join Relationships

- One-to-many relationships include the Primary Key of the ‘one’ table and a Foreign Key (FK) in the ‘many’ table.

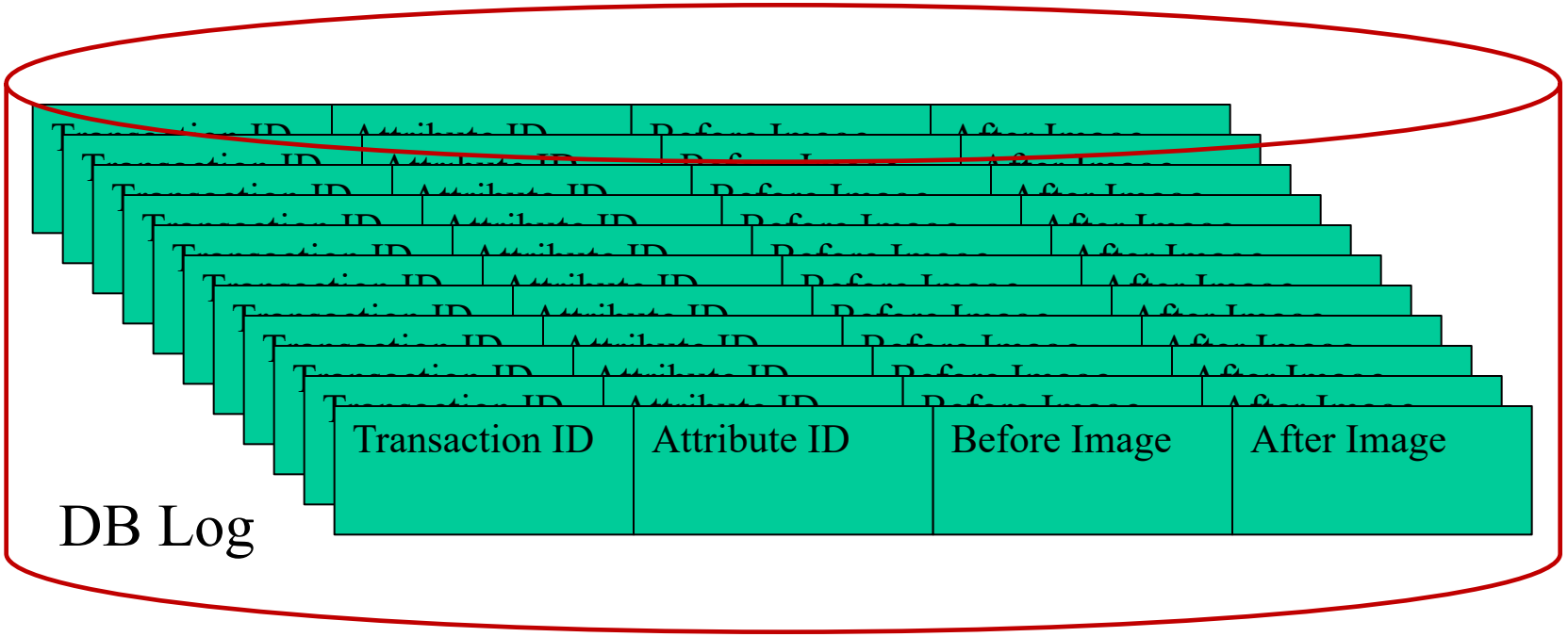
Ensuring Database Integrity

- Database integrity involves the maintenance of the logical and business rules of the database.
- There are two kinds of “DB Integrity” that must be addressed:
  - Entity Integrity
  - Referential Integrity

Strategies for managing Integrity

- You could ignore it, but it costs you time.
- Place the Burden on your customer or user.
- Have the programmers “fix the problem”
- Place the burden on the Database Management System (DBMS)
- Temporal integrity is one of the key challenges of Address Database management.

Database Recovery – Rollback OA



Rollback

53	Table: EMP, Key=XXXXX, Attr: BDATE	01/01/1997	03/09/1996
Transaction ID	Attribute ID	Before Image	After Image

For Transaction 53: Replace for Row=XXXXX, BDATE with **Before Image**

Using SQL- Structured Query Language

- SQL is a standard database protocol, adopted by most ‘relational’ databases
- Provides syntax for data:
  - Definition
  - Retrieval
  - Functions (COUNT, SUM, MIN, MAX, etc)
  - Updates and Deletes

Data Retrieval

- **SELECT** list **FROM** table **WHERE** condition
- list - a list of items or * for all items
  - WHERE - a logical expression limiting the number of records selected
  - can be combined with Boolean logic: AND, OR, NOT
  - ORDER may be used to format results

UPDATE tables

- SET item = expression
- WHERE expression
- INSERT INTO table
- VALUES …..

Step1: Creating a Data Model

- Identify Candidate Entities
- Identify Relationships
- Define Entities & Relationships
- Review Entity-Relationship Model

Step 2: Defining an Attribute Model

- List Candidate Attributes for each Entity
- Add KEYS to model
- Attribute & Normalize Model
- Define Attributes
- Review Logical Model

Step 3: Identify & Capture Business Rules

- Review & Verify Cardinalities
- Define Referential Integrity
- Identify Business Domains
- Identify Attribute Default Values

Step 4: Define Physical Model

- Select Target DBMS
- Name Tables & Columns
- Name & Define Indexes
- Define Columns
- Verify/Update Triggers
- Generate Reports & Document Design

Step 5: Review Final Design

- Verify Entities & Definitions
- Verify Relationships & Definitions
- Verify Attributes & Definitions
- Verify Business Constraints
- Approve Schema Design

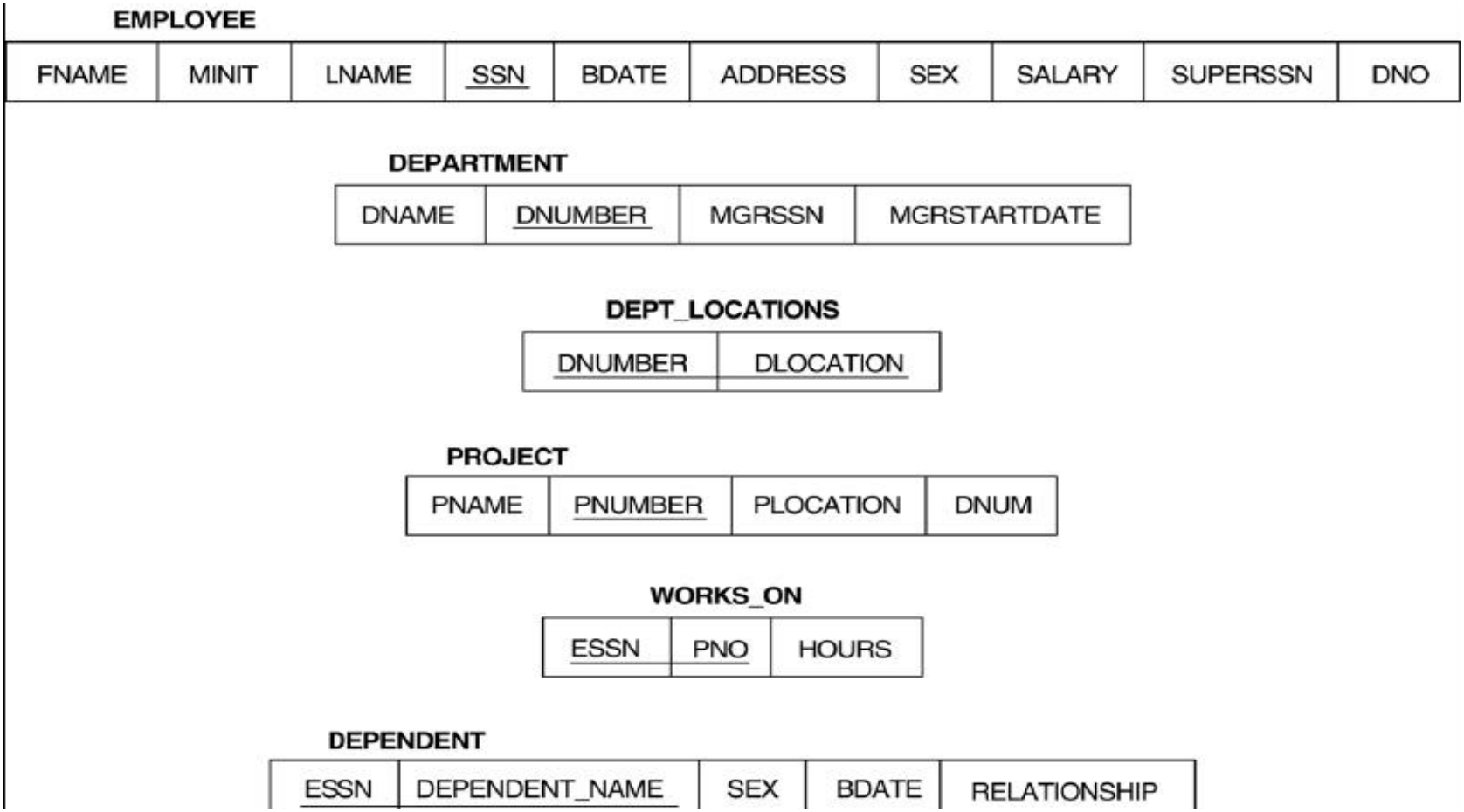
Retrieval Queries in SQL

Basic form of the SQL SELECT statement is called a *mapping* or a *SELECT-FROM-WHERE block*

SELECT <attribute list>
FROM <table list>
WHERE <condition>

– <attribute list> is a list of attribute names whose values are to be retrieved by the query
– <table list> is a list of the relation names required to process the query
– <condition> is a conditional (Boolean) expression that identifies the tuples to be retrieved by the query

Relational Database schema



Populated Database:

EMPLOYEE	FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
	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	1968-07-19	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

DEPT_LOCATIONS	DNUMBER	DLOCATION
	1	Houston
	4	Stafford
	5	Bellaire
	5	Sugarland
	5	Houston

DEPARTMENT	DNAME	DNUMBER	MGRSSN	MGRSTARTDATE
	Research	5	333445555	1988-05-22
	Administration	4	987654321	1995-01-01
	Headquarters	1	888665555	1981-06-19

WORKS_ON	ESSN	PNO	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	null

PROJECT	PNAME	PNUMBER	PLOCATION	DNUM
	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	ESSN	DEPENDENT_NAME	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

Summary of SQL Queries

- A query in SQL can consist of up to six clauses, but only the first two, SELECT and FROM, are mandatory. The clauses are specified in the following order:

SELECT <attribute list>
FROM <table list>
[WHERE <condition>]
[GROUP BY <grouping attribute(s)>]
[HAVING <group condition>]
[ORDER BY <attribute list>]

Summary of SQL Queries (cont.)

- The SELECT-clause lists the attributes or functions to be retrieved
- The FROM-clause specifies all relations (or aliases) needed in the query but not those needed in nested queries
- The WHERE-clause specifies the conditions for selection and join of tuples from the relations specified in the FROM-clause
- GROUP BY specifies grouping attributes
- HAVING specifies a condition for selection of groups
- ORDER BY specifies an order for displaying the result of a query
- A query is evaluated by first applying the WHERE-clause, then
- GROUP BY and HAVING, and finally the SELECT-clause

Simple Queries

Query 0: Retrieve the birthdate and address of the employee whose name is 'John B. Smith'.

```
Q0:   SELECT BDATE, ADDRESS
      FROM EMPLOYEE
      WHERE FNAME='John' AND MINIT='B'
            AND LNAME='Smith'
```

Query 1: Retrieve the name and address of all employees who work for the 'Research' department.

```
Q1:   SELECT  FNAME, LNAME, ADDRESS
      FROM EMPLOYEE, DEPARTMENT
      WHERE  DNAME='Research' AND
            DNUMBER=DNO
```

Some Queries Cont.

Query 2: For every project located in 'Stafford', list the project number, the controlling department number, and the department manager's last name, address, and birthdate.

Q2:
```
SELECT PNUMBER, DNUM, LNAME, BDATE, ADDRESS
FROM PROJECT, DEPARTMENT, EMPLOYEE
WHERE DNUM=DNUMBER AND MGRSSN=SSN AND
PLOCATION='Stafford'
```

Query 3: For each employee, retrieve the employee's name, and the name of his or her immediate supervisor.

Q3:
```
SELECT E.FNAME, E.LNAME, S.FNAME, S.LNAME
FROM EMPLOYEE E S
WHERE E.SUPERSSN=S.SSN
```

Some Queries Cont. JOIN

QT: **SELECT E.FNAME, E.LNAME, S.FNAME, S.LNAME**
FROM EMPLOYEE E S
WHERE E.SUPERSSN=S.SSN

Can be written as:

QTA: **SELECT E.FNAME, E.LNAME, S.FNAME, S.LNAME**
FROM (EMPLOYEE E LEFT OUTER JOIN EMPLOYEES
ON E.SUPERSSN=S.SSN)

Some Queries Cont. JOIN

Q9:
```
SELECT FNAME, LNAME, ADDRESS
FROM EMPLOYEE, DEPARTMENT
WHERE DNAME='Research' AND DNUMBER=DNO
```

Can be written as:

Q9A:
```
SELECT FNAME, LNAME, ADDRESS
FROM (EMPLOYEE JOIN DEPARTMENT
ON DNUMBER=DNO)
WHERE DNAME='Research'
```

Or as:

Q9B:
```
SELECT FNAME, LNAME, ADDRESS
FROM (EMPLOYEE NATURAL JOIN
DEPARTMENT AS DEPT(DNAME, DNO, MSSN, MSDATE)
WHERE DNAME='Research'
```

AGGREGATE FUNCTIONS

Include **COUNT**, **SUM**, **MAX**, **MIN**, and **AVG**

Query 10: Find the maximum salary, the minimum salary, and the average salary among all employees.

```
Q10:    SELECT MAX(SALARY),  MIN(SALARY), AVG(SALARY)
        FROM EMPLOYEE
```

Query 11: Find the maximum salary, the minimum salary, and the average salary among employees who work for the 'Research' department.

```
Q11:    SELECT MAX(SALARY), MIN(SALARY), AVG(SALARY)
        FROM EMPLOYEE, DEPARTMENT
        WHERE DNO=DNUMBER AND DNAME='Research'
```

Compute

Row aggregate function	Result
AVG	Average of the values in the numeric expression
COUNT	Number of selected rows
MAX	Highest value in the expression
MIN	Lowest value in the expression
STDEV	Statistical standard deviation for all values in the expression
STDEVP	|Statistical standard deviation for the population for all values in the expression
SUM	Total of the values in the numeric expression
VAR	Statistical variance for all values in the expression
VARP	Statistical variance for the population for all values in the expression

ACID

ACID is an acronym that stands for Atomicity, Consistency, Isolation, Durability. These are explained below.

Atomicity

Atomicity means that you guarantee that either all of the transaction succeeds or none of it does. You don't get part of it succeeding and part of it not. If one part of the transaction fails, the whole transaction fails. With atomicity, it's either "all or nothing".

Consistency

This ensures that you guarantee that all data will be consistent. All data will be valid according to all defined rules, including any constraints, cascades, and triggers that have been applied on the database.

Isolation

Guarantees that all transactions will occur in isolation. No transaction will be affected by any other transaction. So a transaction cannot read data from any other transaction that has not yet completed.

Durability

Durability means that, once a transaction is committed, it will remain in the system – even if there's a system crash immediately following the transaction. Any changes from the transaction must be stored permanently. If the system tells the user that the transaction has succeeded, the transaction must have, in fact, succeeded.

When is ACID needed?

- ACID properties are designed as principles of transaction-oriented database recovery.

- ACID provides the principles that database transactions should adhere to, to ensure that data doesn't become corrupt as a result of a failure of some sort.

- A transaction is a single logical operation that may consist of one or many steps. For example, transferring money between bankaccounts (i.e. debiting one account and crediting the other) is a transaction.

- If a transaction like this fails halfway through, it could have major consequences. Money could be debited from the first account but not credited to the other account.

When is ACID needed?

- According to the ACID definition, a database is consistent if and only if it contains the results of successful transactions.
- Any database that is ACID-compliant will ensure that only successful transactions are processed.
- If a failure occurs before a transaction completes, no data will be changed.

ACID-compliant DBMSs provide organisations with the confidence that their database will maintain data integrity, even if some type of failure occurs while transactions are in the middle of being processed.

But what is the cost associated to be ACID compliant?