Databases and Advanced Data Techniques

Tarapong Sreenuch

1 Relational Data Model and SQL Query Formulation

Lecture Outline

- Database Management Systems (DBMS)
- Entity-Relationship Model
- SQL and Non-Procedural Access
- SELECT and JOIN Statements
- Integrity Rules
- Row Summary and GROUP BY
- Wrapping Up

Database Management System

major part of software industry

revolutionary evolvement over 40 years

foundation for management of long-term memory of organizations

DATA

Sources of Data

Conventional Facts

Unconventional Facts

Data Characteristics

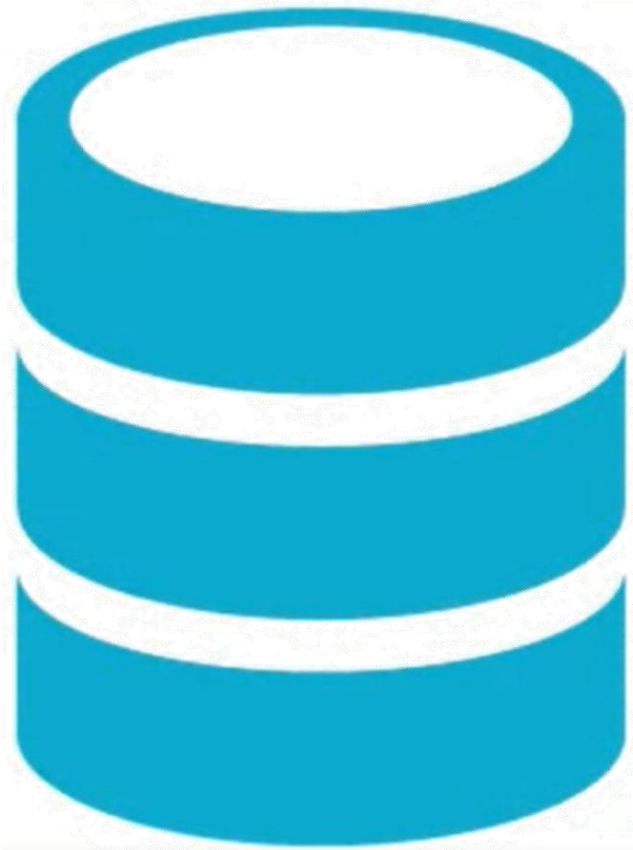
Inter-related

Persistency

Shared

Database Application Examples

University Database



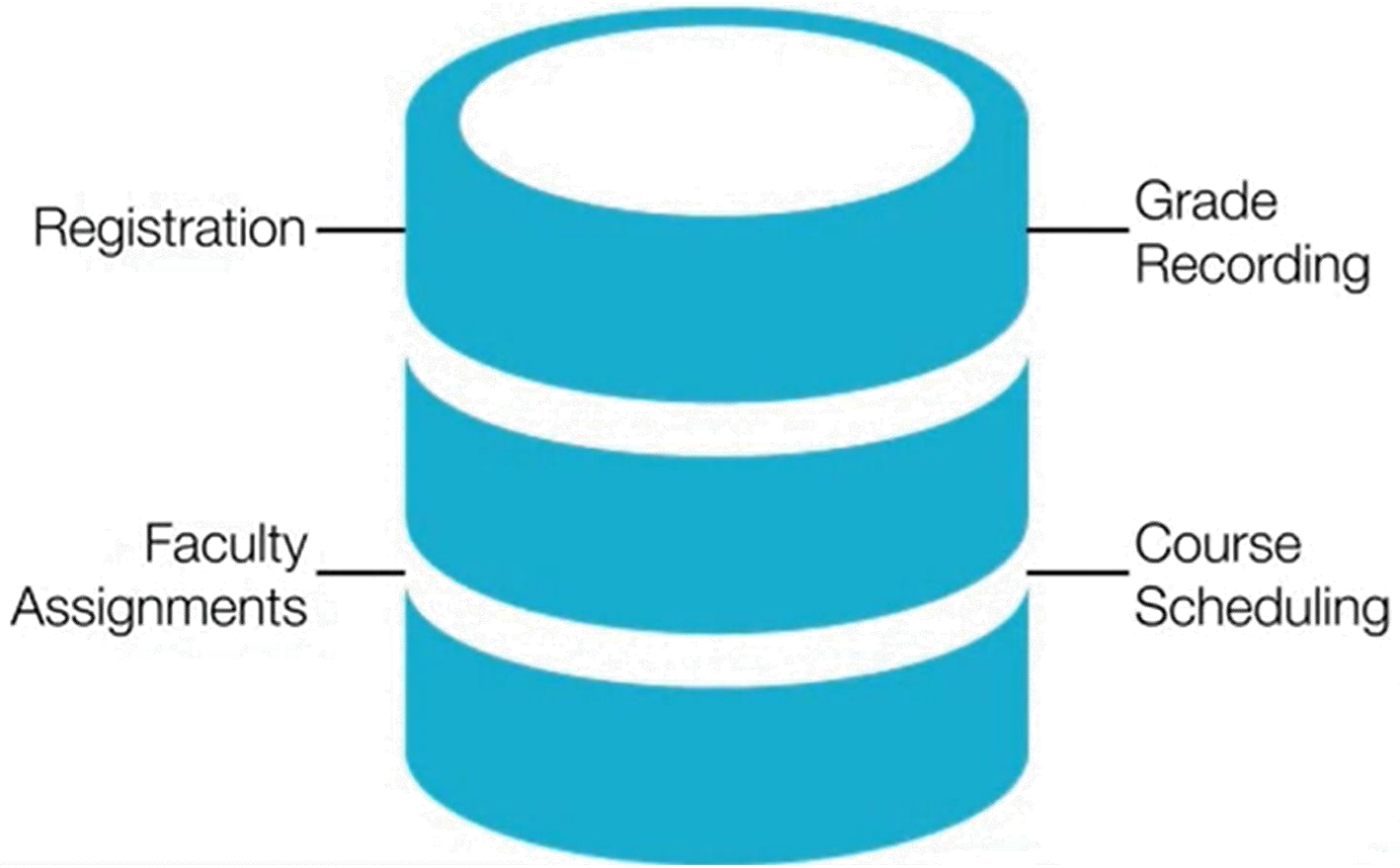
Entities:

- Students
- Faculty
- Courses
- Course offerings
- Enrollments

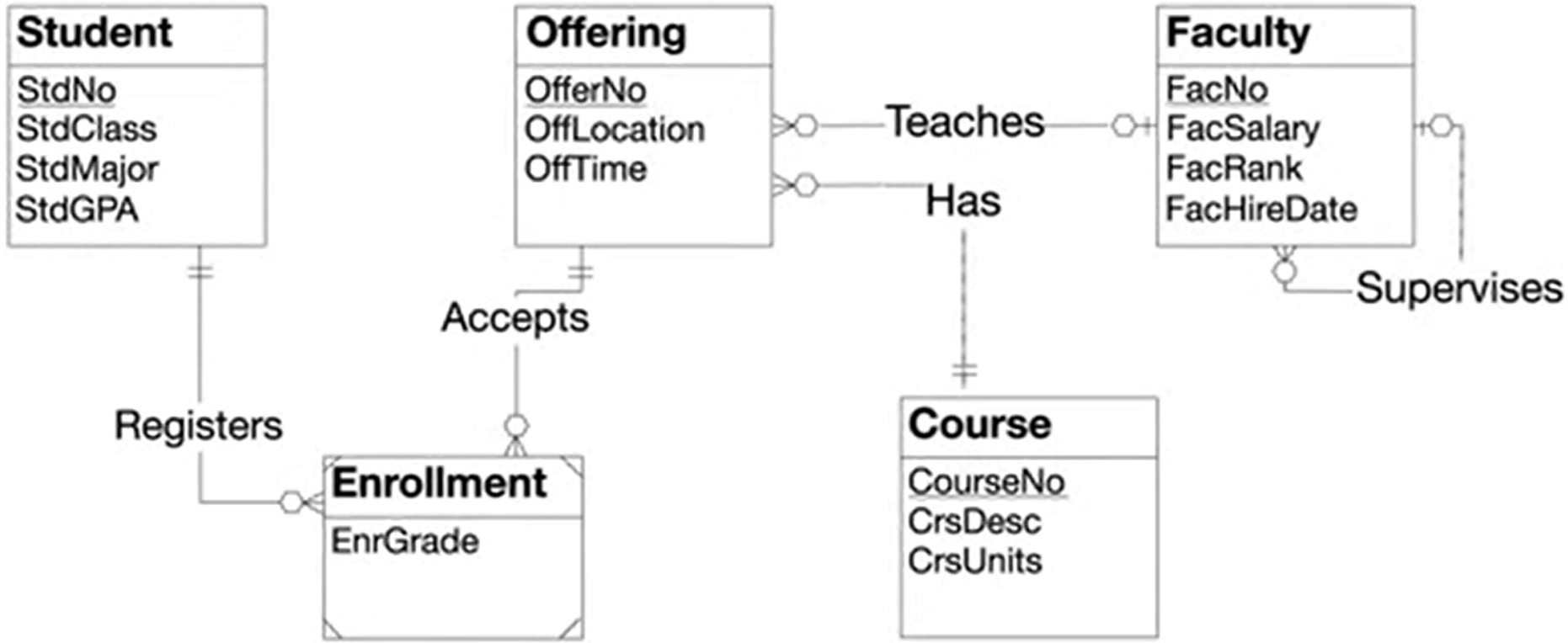
Relationships:

- Faculty Course Assignments
- Students Enrollment
- Course Scheduling

Operational Usages



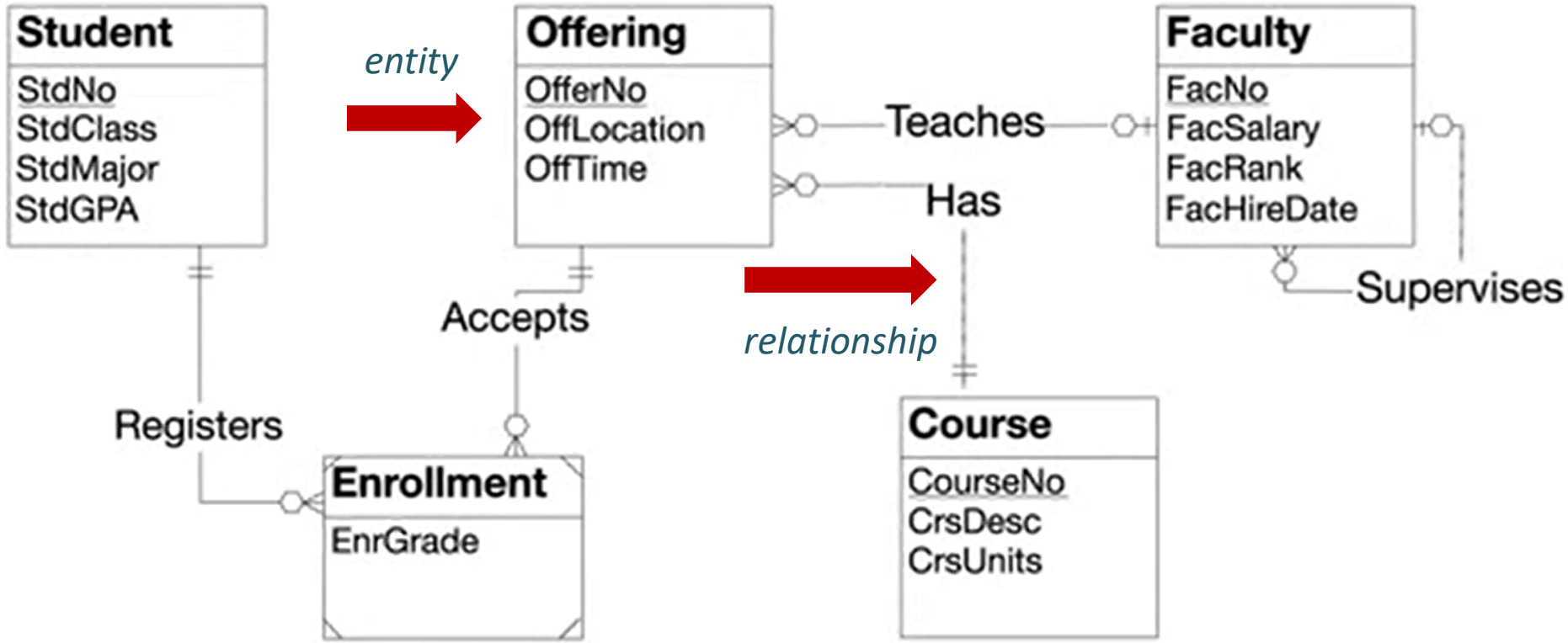
ERD – Entity Relationship Diagram



Entity Relationship Diagram

An entity-relationship diagram (ERD) is a type of data modeling that shows a graphical representation of objects or concepts within an information system or organization and their relationship to one another.

ERD – Entity Relationship Diagram



Overview of Databases

ENTERPRISE DBMS's

Organizations with very large databases
Thousands of simultaneous users
Strong reliability & performance

DESKTOP DBMS's

Small workgroups with smaller databases
Tens of simultaneous users
Modest reliability & performance

EMBEDDED DBMS's

Sold by value-added software resellers
Hidden from users
Require little to no ongoing maintenance

Enterprise Databases

Desktop Application (How It's Different From Databases)

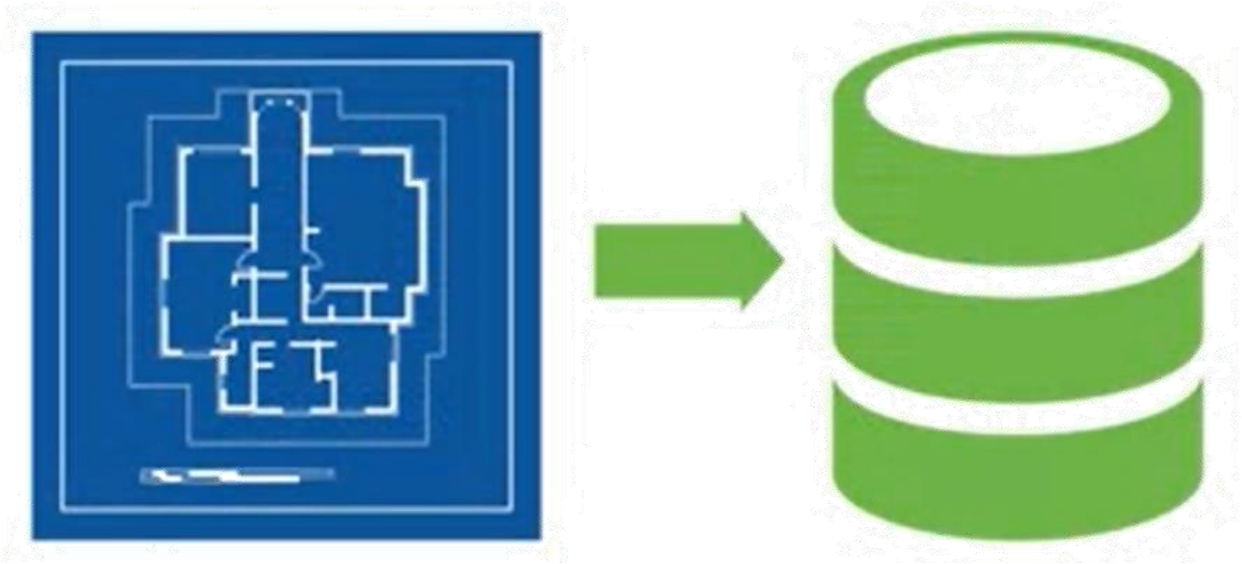
Little or no planning is required for Spreadsheets and Documents.

Spreadsheet

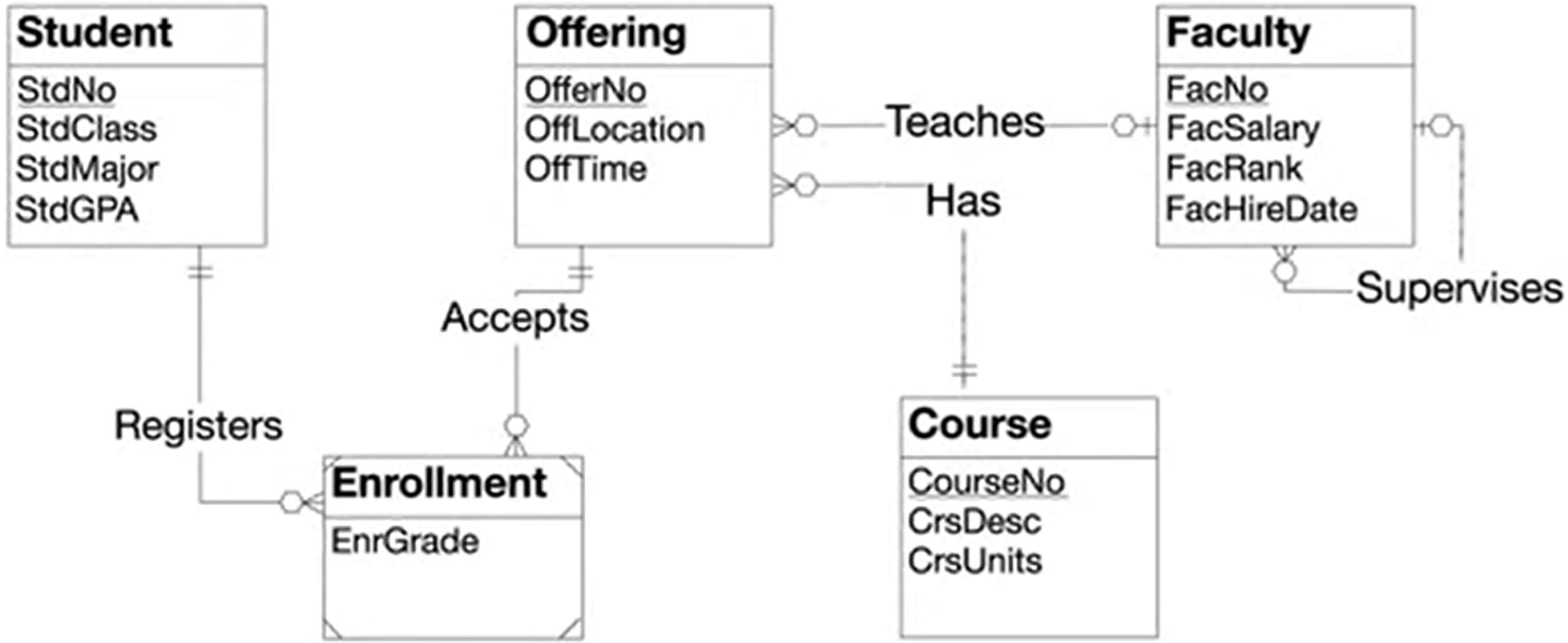
Document

Planning for Databases

Planning is essential for databases,
even for small workgroup databases.



Entity Relationship Diagram



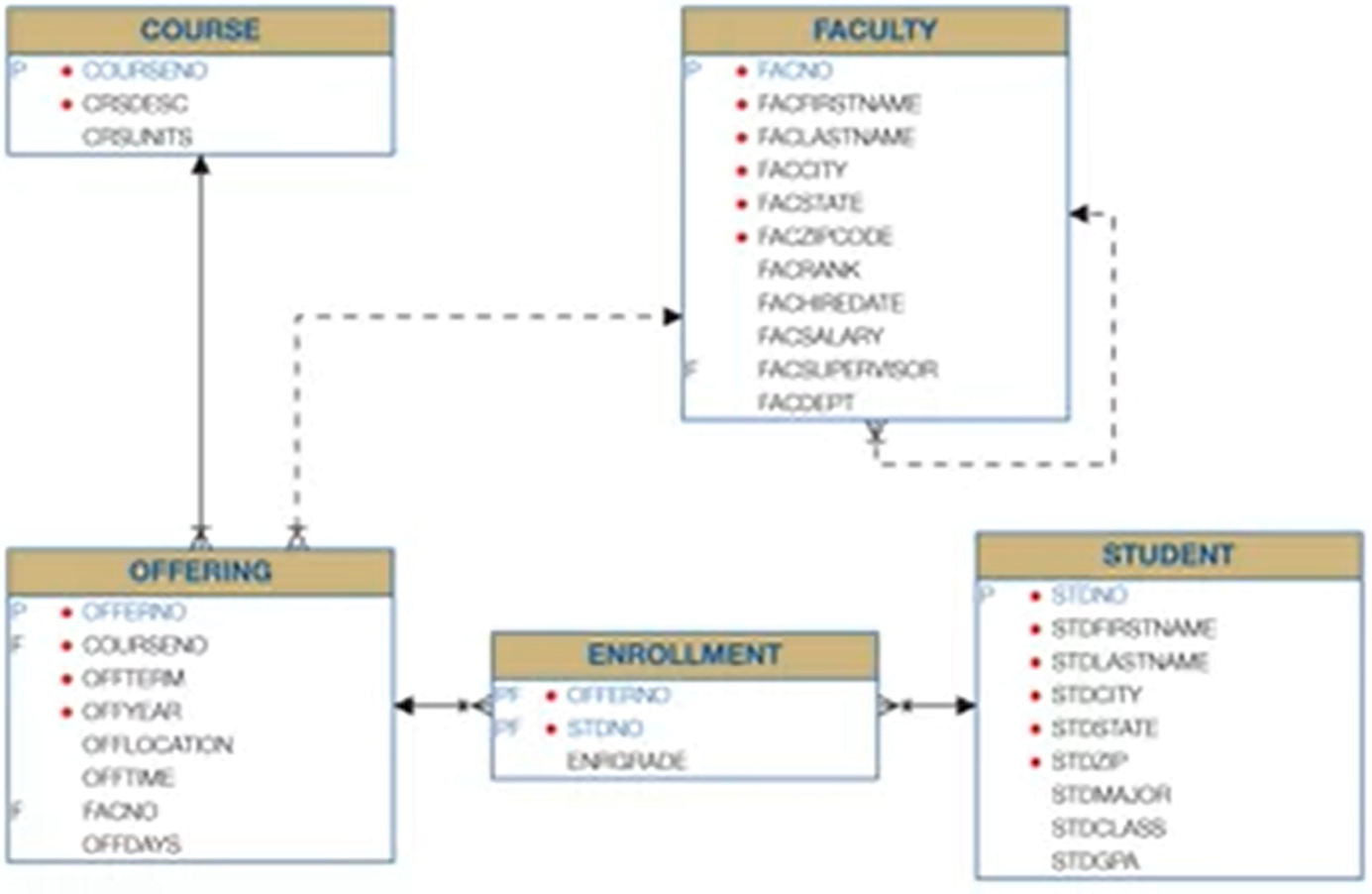
Structured Query Language (SQL)

SQL is used to communicate with a database.
It is the standard language for relational database management systems.

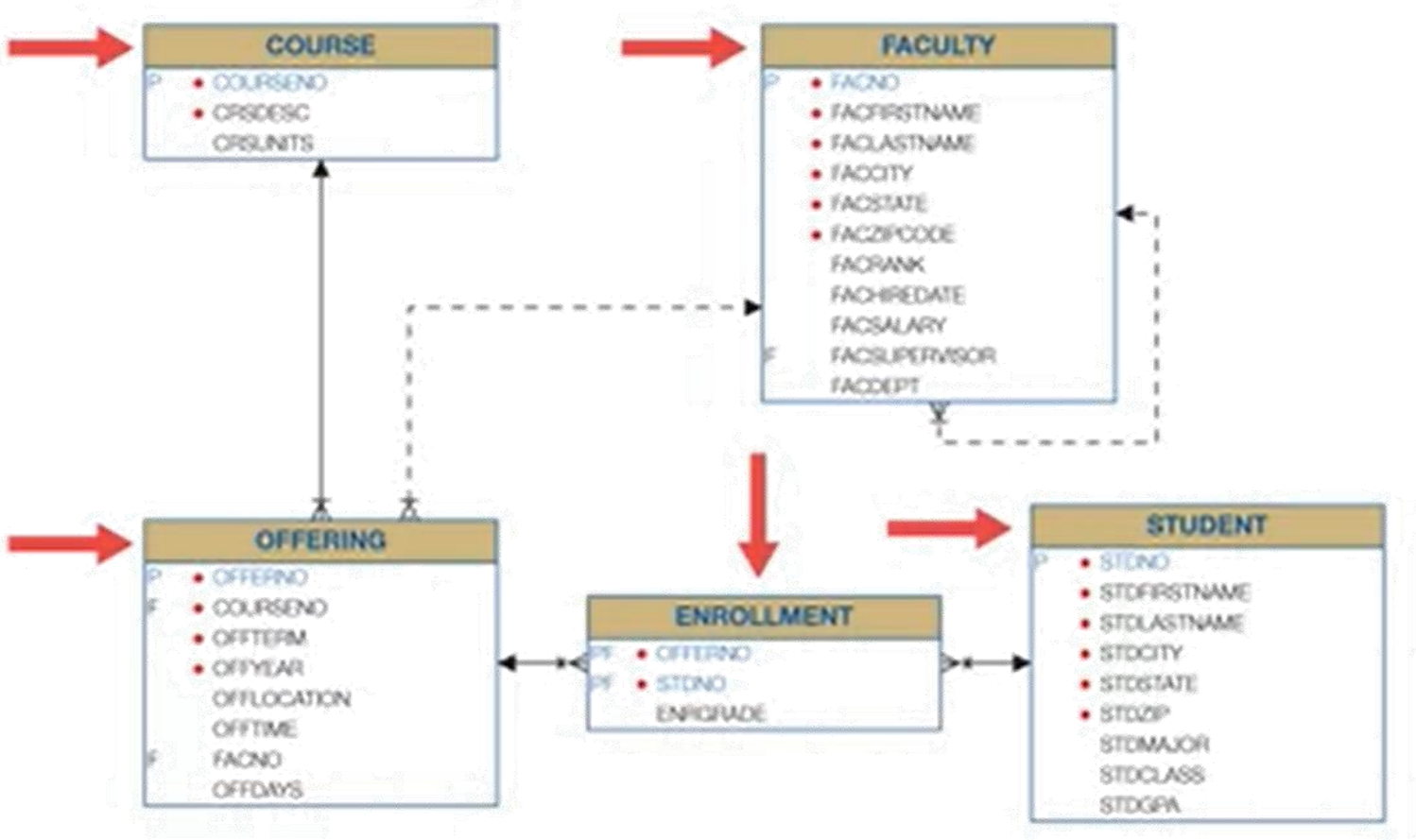
Example:

```
CREATE TABLE table_name
(
column_name1 data_type(size),
column_name2 data_type(size),
column_name3 data_type(size),
....
);
```

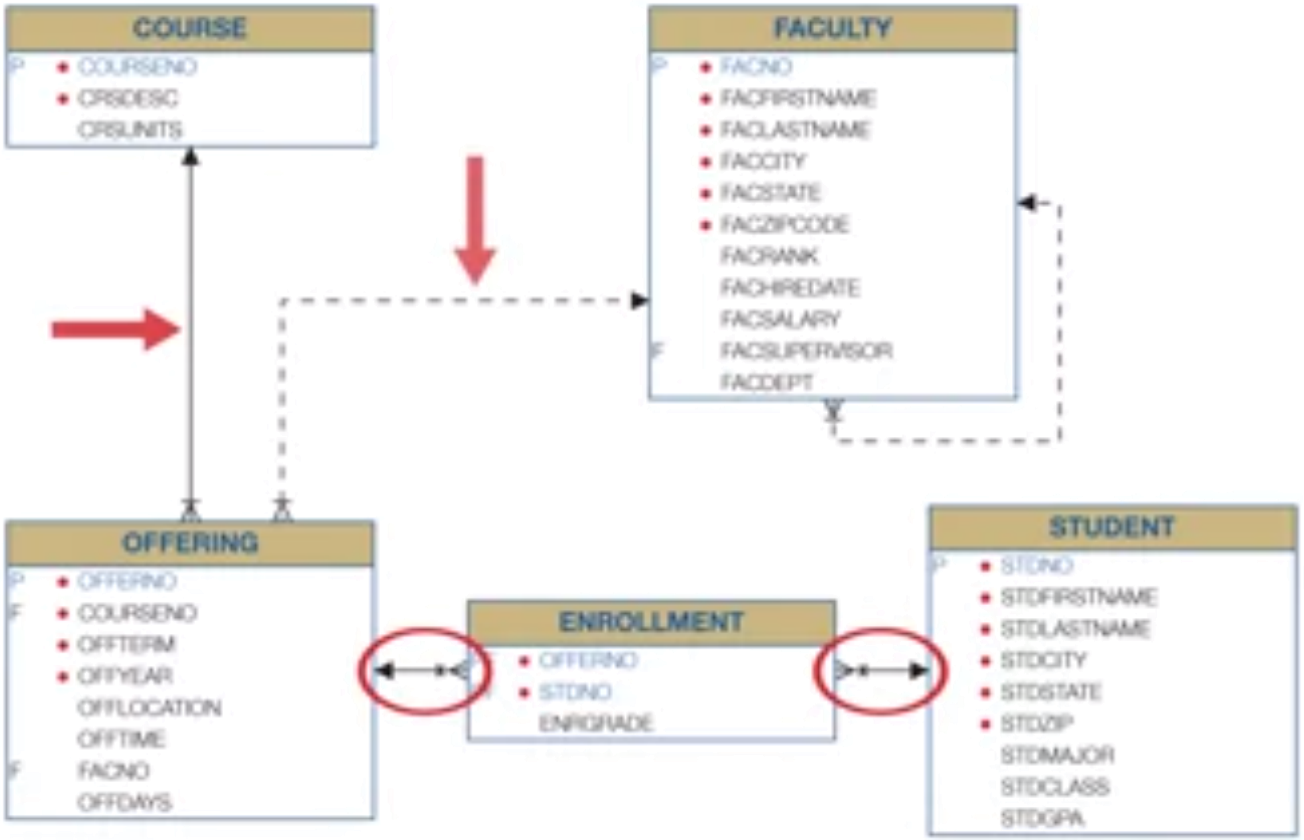
Oracle Relational Diagram



Relational Diagram: Entity



Relational Diagram: Relationship



Database Management System

A collection of components that supports the creation, use and maintenance of databases.

Non-Procedural Access

Set-Oriented

We only need to specify what data to retrieve rather than how to retrieve it.

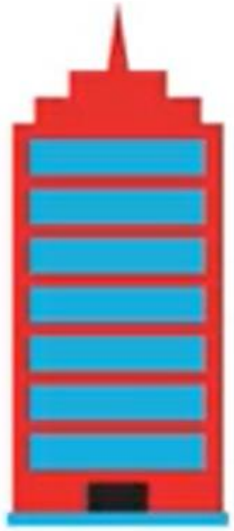
Query

A database query can be either a simple data retrieval query or an action query that performs additional operations on the data, such as insertion, updating or deletion.

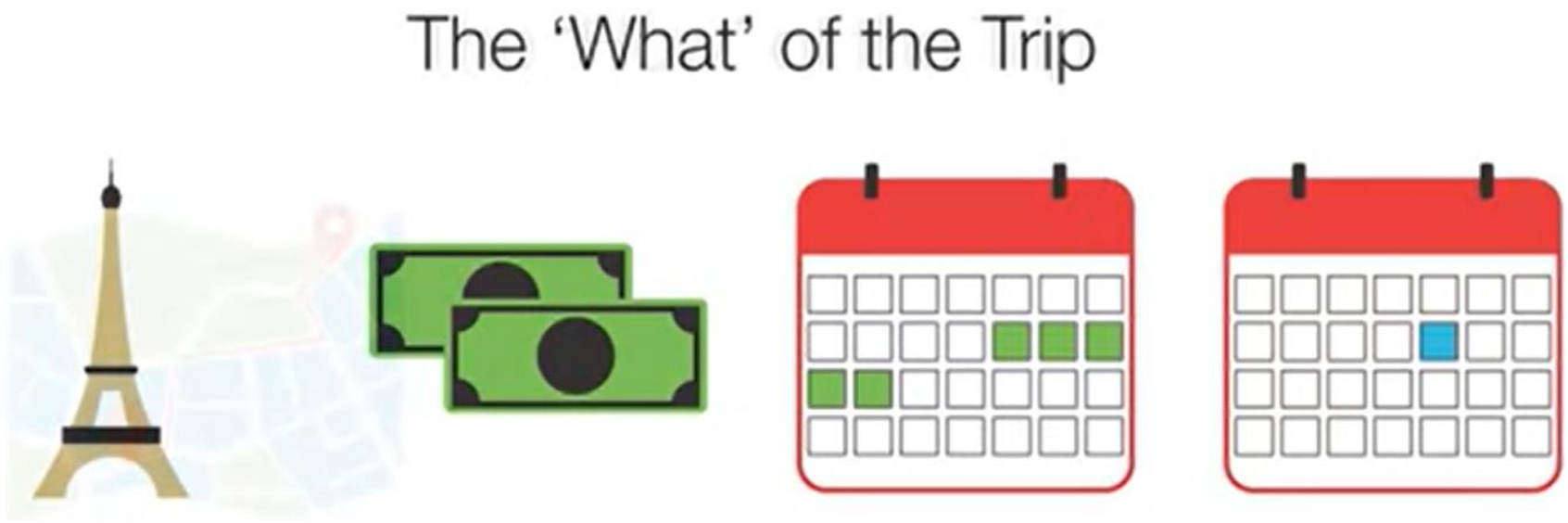
SQL Query

Planning A Vacation: How?

The ‘How’ of the Trip



Planning A Vacation: What?



Planning A Vacation: SQL Query

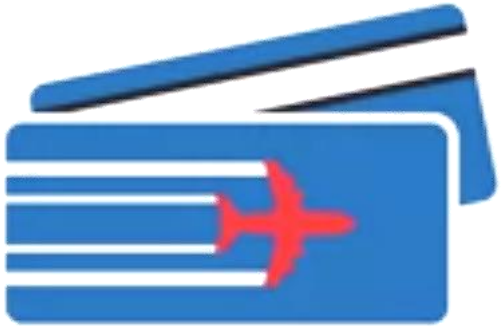
Sample SELECT Statement

```
SELECT OfferNo, CourseNo, FacFirstName, FacLastName
 FROM Offering INNER JOIN Faculty
     ON Faculty.FacNo = Offering.FacNo
WHERE OffTerm = 'FALL' AND OffYear = 2016
    AND FacRank = 'ASST' AND CourseNo LIKE 'IS%';
```

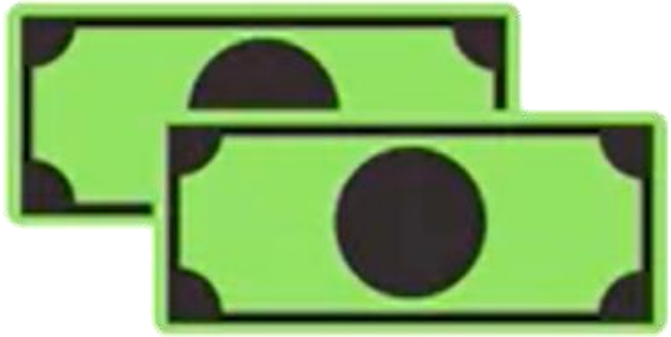
Sample SELECT Statement

```
SELECT OfferNo, CourseNo, FacFirstName, FacLastName
 FROM Offering INNER JOIN Faculty
     ON Faculty.FacNo = Offering.FacNo
WHERE OffTerm = 'FALL' AND OffYear = 2016
    AND FacRank = 'ASST' AND CourseNo LIKE 'IS%';
```

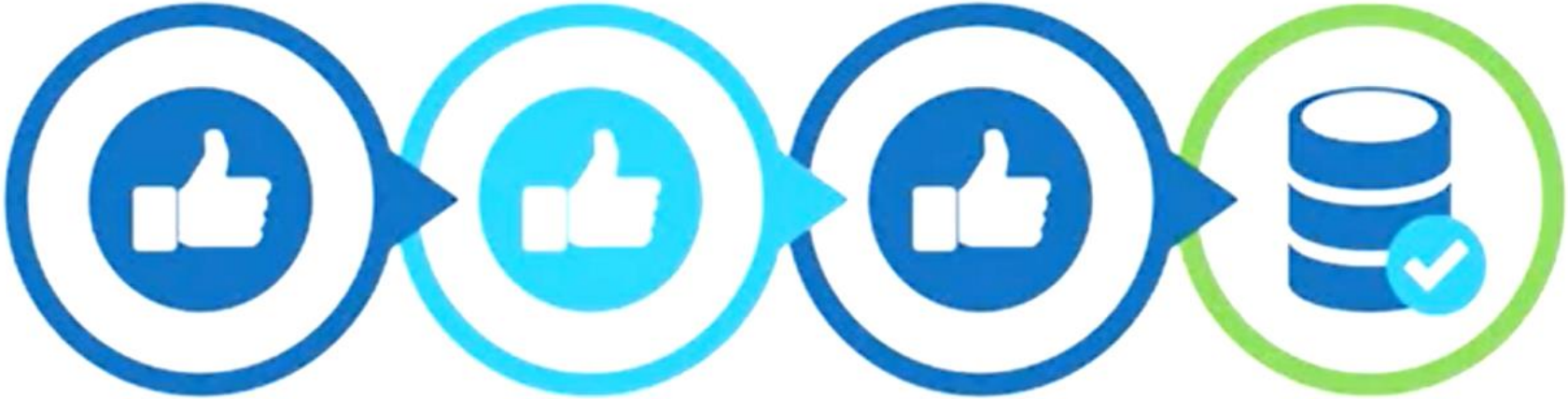
Transaction Management



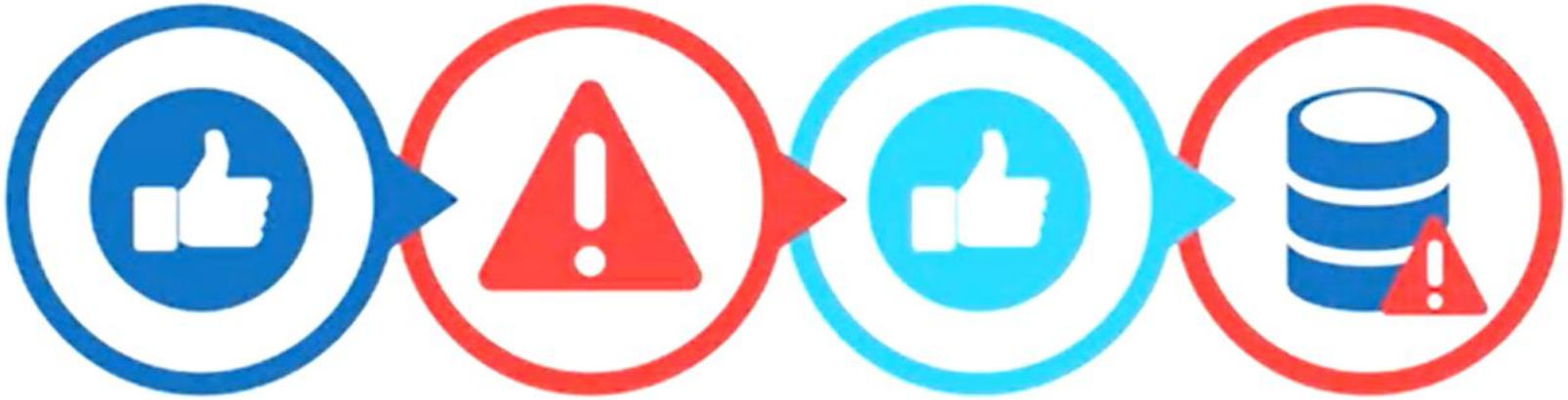
Buying A Car



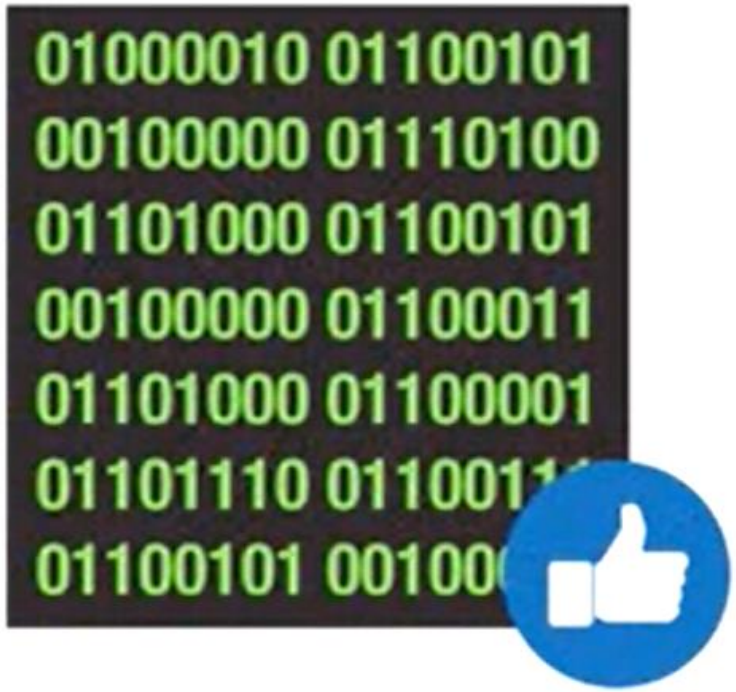
Database Transactions



Database Transactions



No Loss Of Data Requirement



Airline Transaction Example

```
START TRANSACTION
    Display greeting
    Get reservation preferences from user
    SELECT departure and return flight records
    If reservation is acceptable then
        UPDATE seats remaining of departure flight record
        UPDATE seats remaining of return flight record
        INSERT reservation record
        Email receipt if requested
    End If
    On Error: ROLLBACK
COMMIT
```

ATM Transaction Example

```
START TRANSACTION
    Display greeting
    Get account number, pin, type, and amount
    SELECT account number, type, and balance
    If balance is sufficient then
        UPDATE account by posting debit
        UPDATE account by posting credit
        INSERT history record
        Display message and dispense cash
        Print receipt if requested
    End If
    On Error: ROLLBACK
COMMIT
```

Transaction Processing

- Supports daily operations of an organisation
- Collection of database operations
- Reliable and efficient processing of transactions as one unit of work
- No lost data due to interference among multiple users
- Recover from failures without loss of data for completed transactions

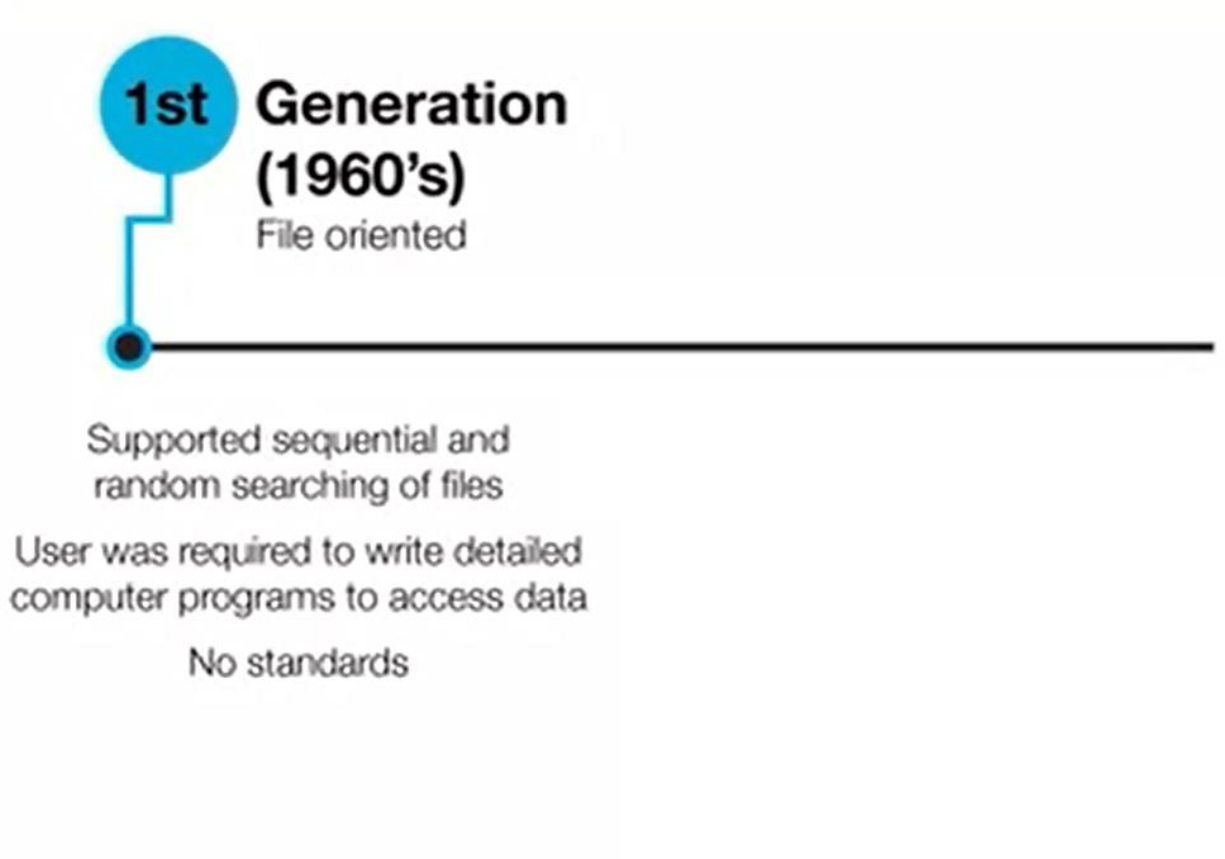
DBMS Internal Features

Concurrency Control Manager

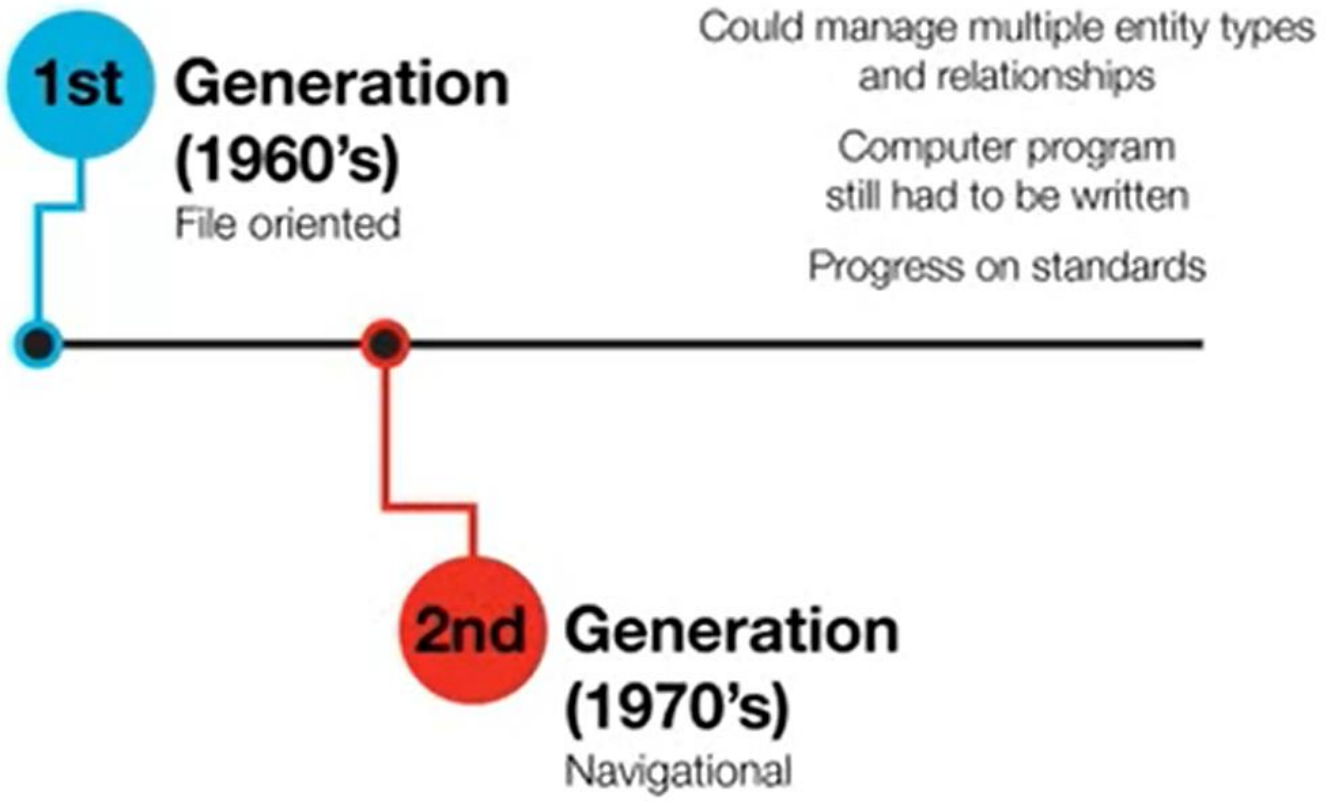
Recovery Manager

Transparent Services for Application Developers

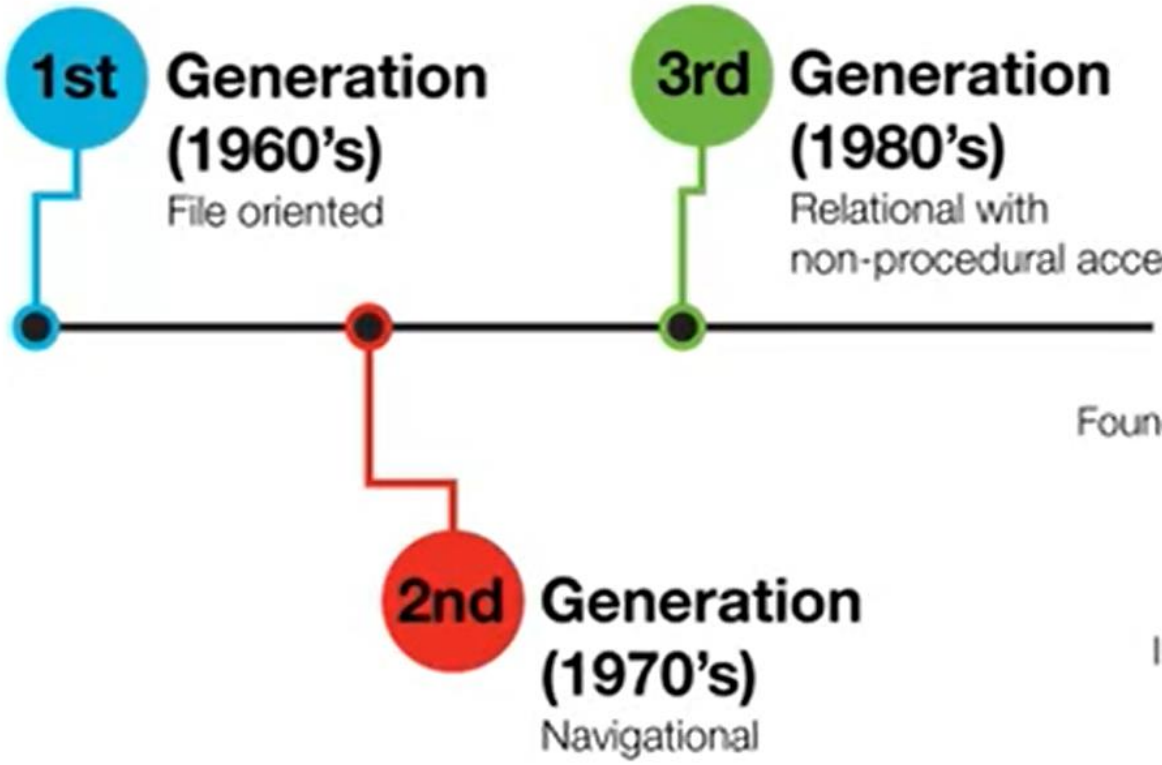
DBMS Technology Evolution



DBMS Technology Evolution



DBMS Technology Evolution

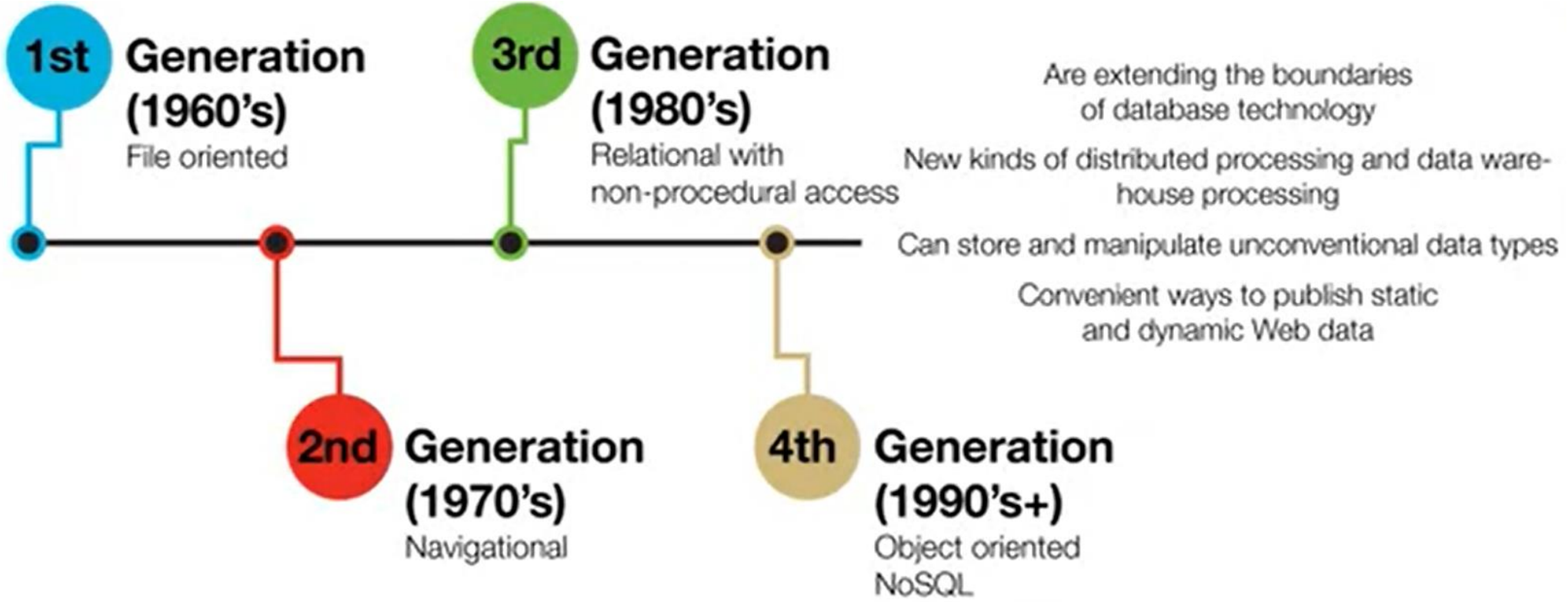


Foundation based on mathematical relations and associated operators.

Optimization technology was developed

IBM performed pioneering research to enable commercialization of relational database technology

DBMS Technology Evolution



What is the consequence of two tax payers or customers with the same government identifier or customer identifier?

What is the consequence of a shipment associated with the wrong order?

Definitions: Null Value

- Absence of a value (missing value)
- Actual value unknown or not applicable for a row

Definitions: Primary Key (PK)

- Column or combination of columns with unique values in each row
- No extraneous columns (minimal)

Sample University Database Tables

Student

StdNo	StdFirstName	StdLastName	StdCity	StdState	StdZip	StdMajor	StdClass	StdGPA
123-45-6789	HOMER	WELLS	SEATTLE	WA	98121-1111	IS	FR	3.00
123-56-7890	BOB	NORBERT	BOTHELL	WA	98011-2121	FIN	JR	2.70
234-56-7890	CANDY	KENDALL	TACOMA	WA	99042-3321	ACCT	JR	3.50

Offering

OfferNo	CourseNo	OffTerm	OffYear	OffLocation	OffTime	FacNo	OffDays
1111	IS320	SUMMER	2013	BLM302	10:30 AM		MW
1234	IS320	FALL	2012	BLM302	10:30 AM	098-76-5432	MW
4321	IS320	FALL	2012	BLM214	3:30 PM	098-76-5432	TTH

Enrollment

OfferNo	StdNo	EnrGrade
1234	123-45-6789	3.3
1234	234-56-7890	3.5
4321	123-45-6789	3.5
4321	124-56-7890	3.2

Definitions: Foreign Key (PK)

- Column or combination of columns
- Related to a primary key in a related table
- Same data type and often same name as related PK

Sample University Database Tables

Student

StdNo	StdFirstName	StdLastName	StdCity	StdState	StdZip	StdMajor	StdClass	StdGPA
123-45-6789	HOMER	WELLS	SEATTLE	WA	98121-1111	IS	FR	3.00
123-56-7890	BOB	NORBERT	BOTHELL	WA	98011-2121	FIN	JR	2.70
234-56-7890	CANDY	KENDALL	TACOMA	WA	99042-3321	ACCT	JR	3.50

Offering

OfferNo	CourseNo	OffTerm	OffYear	OffLocation	OffTime	FacNo	OffDays
1111	IS320	SUMMER	2013	BLM302	10:30 AM		MW
1234	IS320	FALL	2012	BLM302	10:30 AM	098-76-5432	MW
4321	IS320	FALL	2012	BLM214	3:30 PM	098-76-5432	TTH

Enrollment

OfferNo	StdNo	EnrGrade
1234	123-45-6789	3.3
1234	234-56-7890	3.5
4321	123-45-6789	3.5
4321	124-56-7890	3.2

Integrity Rules

Entity Integrity

- Primary key for each table
- No missing (null) values for primary keys
- Ensures traceable entities

Integrity Rules

Referential Integrity

- Two kinds of values for a foreign key in a row
- Match a primary key value of a related table (usual)
- Null value (unusual)
- Ensures valid references among tables

Integrity Rule Violations

Student

StdNo	StdLastName
123-45-6789	WELLS
124-56-7890	KENDALL
234-56-7890	NORBERT
--	JONES

Offering

OfferNo	CourseNo
1234	IS320
4321	IS320

Enrollment

StdNo	OfferNo
123-45-6789	1234
234-56-7890	1234
123-45-6789	4321
124-56-7890	4321
234-56-7890	6789
--	4321

Integrity Rule Violations

Student

StdNo	StdLastName
123-45-6789	WELLS
124-56-7890	KENDALL
234-56-7890	NORBERT
--	JONES

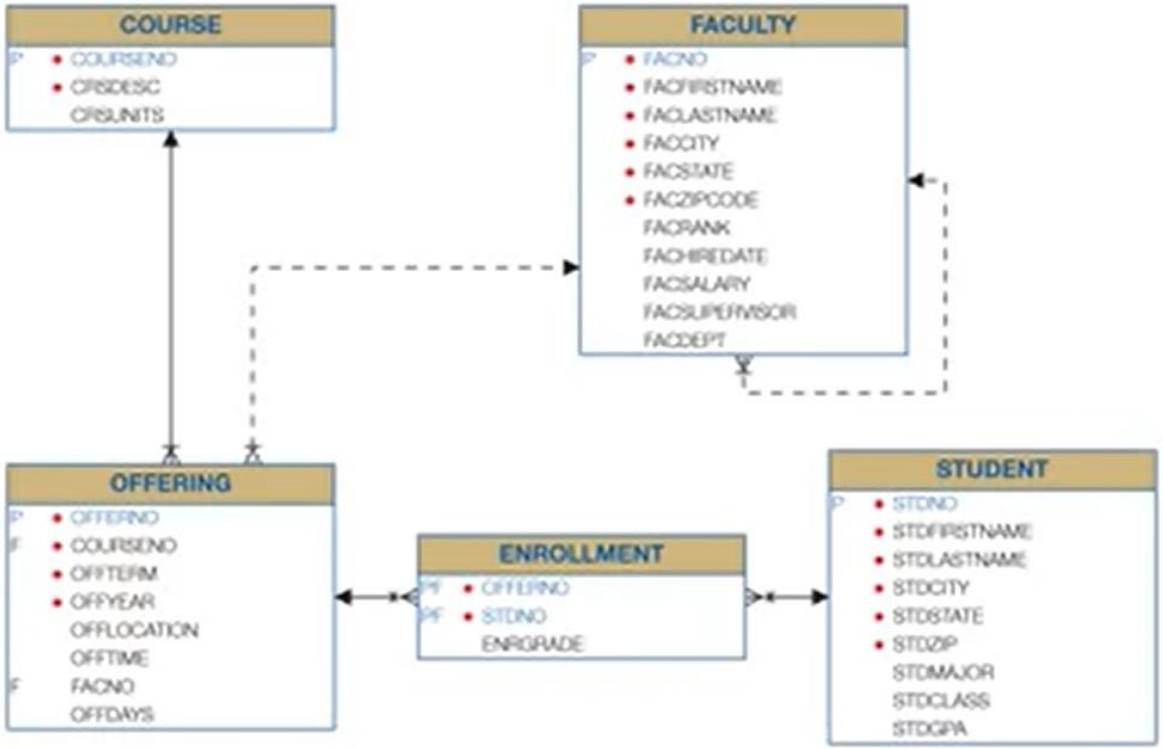
Offering

OfferNo	CourseNo
1234	IS320
4321	IS320

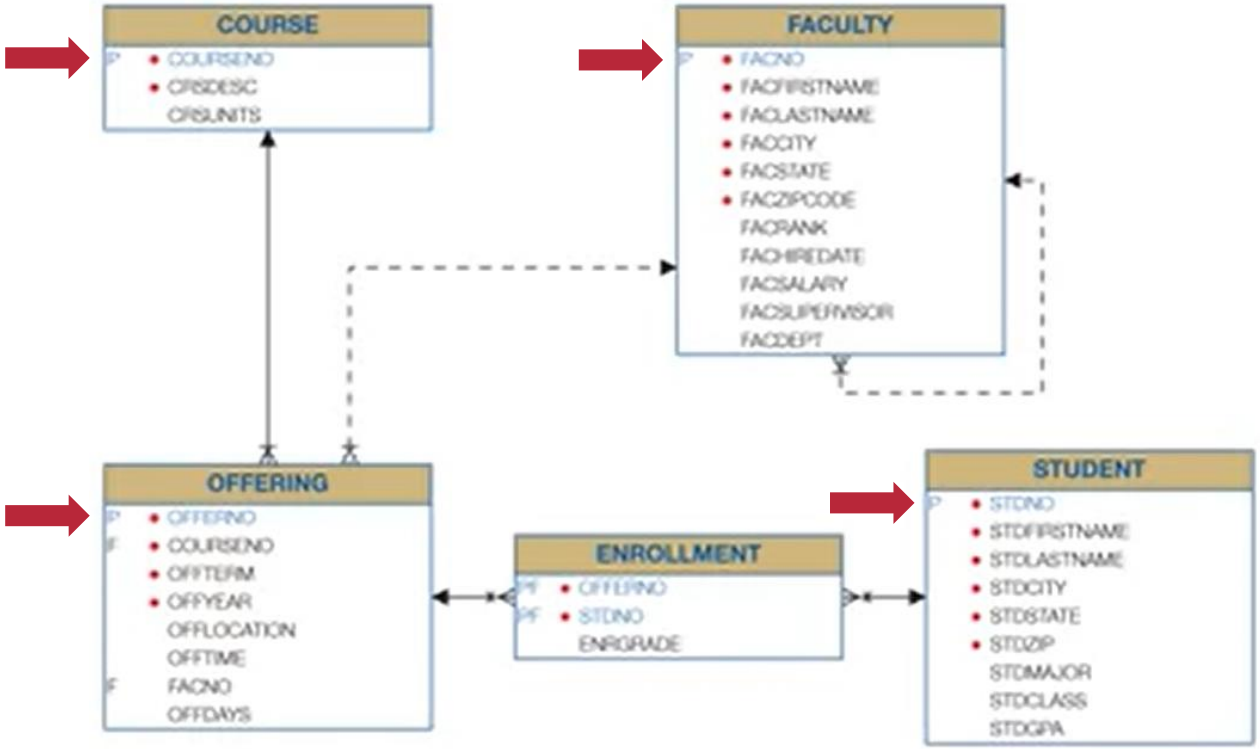
Enrollment

StdNo	OfferNo
123-45-6789	1234
234-56-7890	1234
123-45-6789	4321
124-56-7890	4321
234-56-7890	6789
--	4321

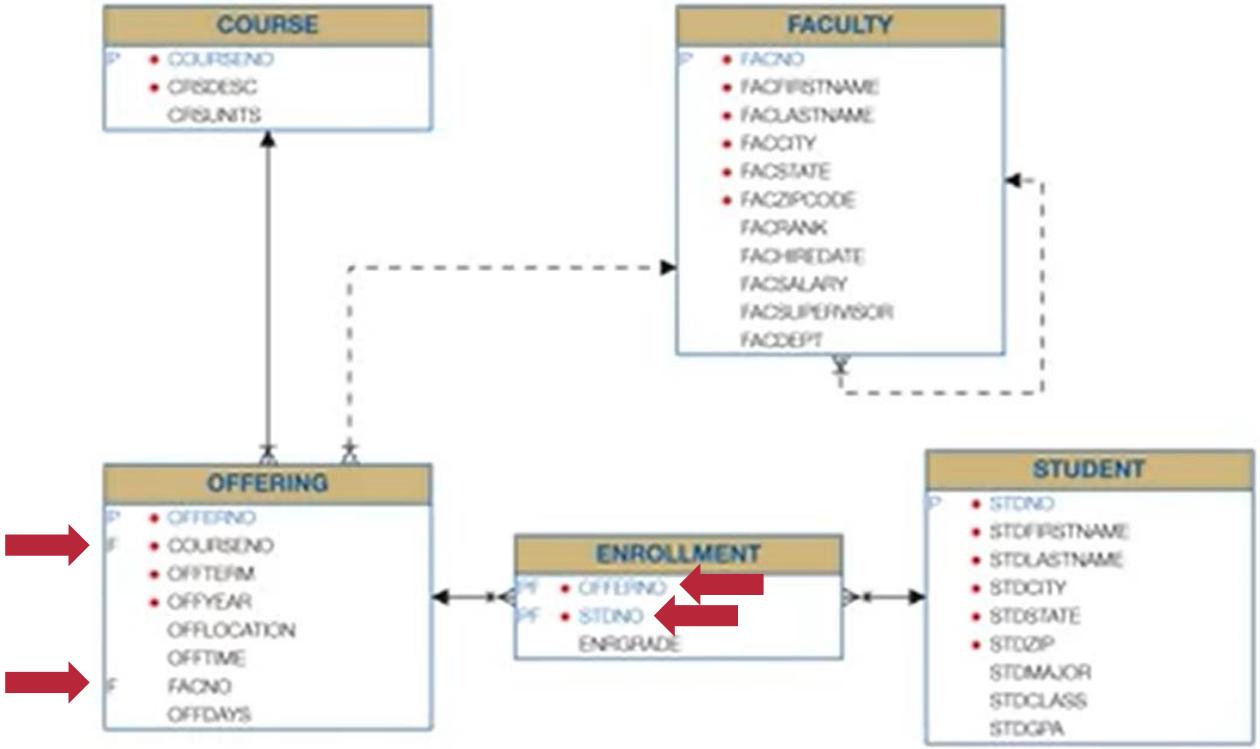
Relational Diagram



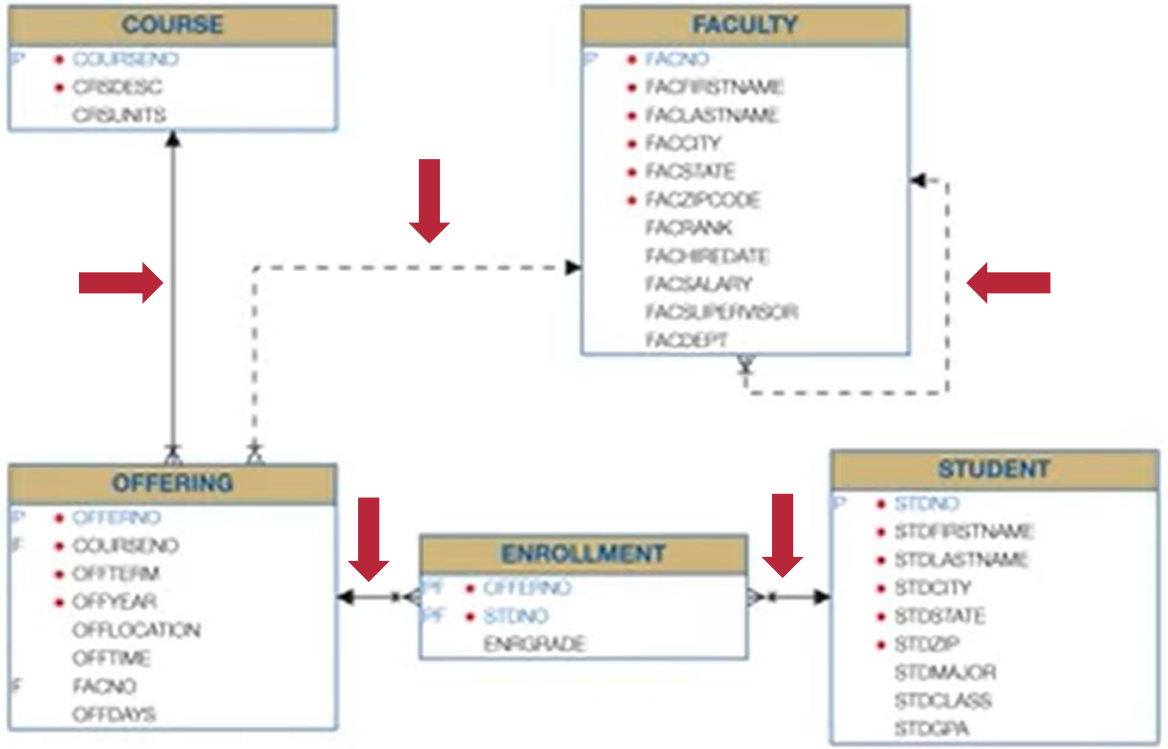
Relational Diagram: Primary Key (PK)



Relational Diagram: Foreign Key (FK)



Relational Diagram: Relationship



Integrity Rules

Entity Integrity: Primary Keys

- Each table has column(s) with unique values
- No missing values for primary keys
- Ensures traceable entities

Integrity Rules

Referential Integrity: Foreign Keys

- Values of a column in one table match values from a source table
- Ensures valid references among tables

Create Table Syntax

- CREATE TABLE <table-name> (<column-list> [<constraint-list>])
- Column list with data types and optional and inline constraints
- Optional external constraint list
  - CONSTRAINT [ConstraintName] <Constraint-Spec>
  - Primary Key
  - Foreign Key
  - Unique
  - Check

Create Table Statement Example

```
CREATE TABLE Student
(  StdNo            CHAR(11),
   StdfFirstName    VARCHAR(50),
   StdLastName      VARCHAR(50),
   StdCity          VARCHAR(50),
   StdState         CHAR(2),
   StdZip           CHAR(10),
   StdMajor         CHAR(6),
   StdClass         CHAR(6),
   StdGPA           DECIMAL(3,2)        )
```

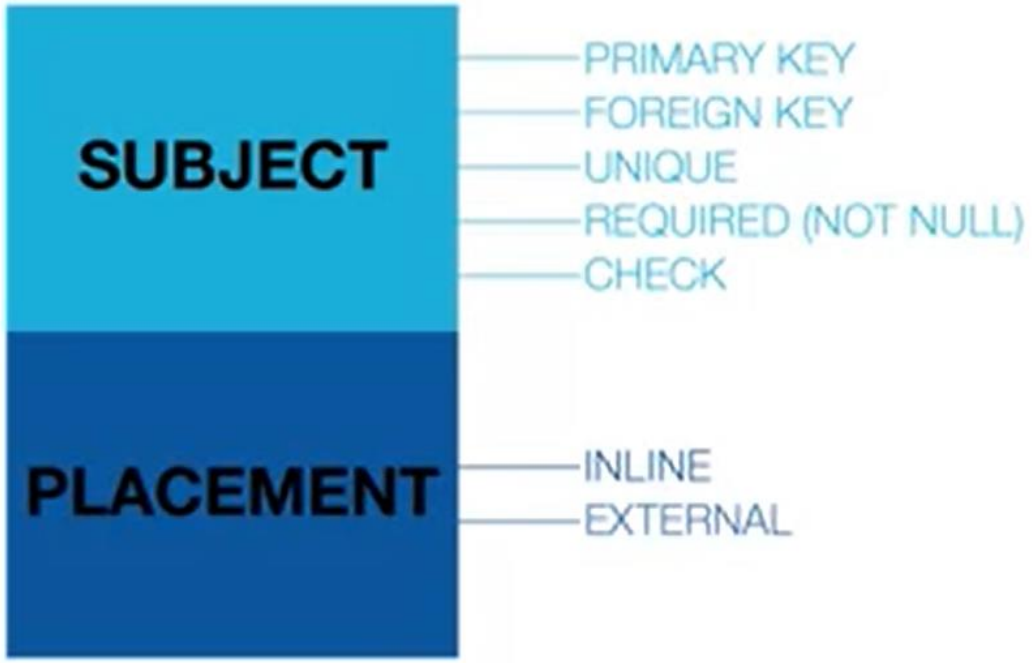
Syntax Error

- Keywords must be spelled exactly.
- Database compilers do not have spelling correction.
- Each column definition is terminated by a comma. A missing comma will result in a syntax error.

Common SQL Data Types

- CHAR(L)
- VARCHAR(L)
- INTEGER
- FLOAT(P)
- DECIMAL(W,R)
- Date/Time: DATE, TIME, TIMESTAMP

Constraint Overview



Constraint Syntax Examples

```
CONSTRAINT  PKCourse  PRIMARY  KEY (CourseNo)
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
CONSTRAINT  PKEnrollment  PRIMARY  KEY
(OfferNo,  StdNo)
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
CONSTRAINT  UniqueCrsDesc  UNIQUE  (CrsDesc)
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
CONSTRAINT  FKOfferNo  FOREIGN  KEY  (OfferNo)
REFERENCES  Offering
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
CONSTRAINT  OffCourseNoReq  NOT  NULL
```

External PK Constraint Placement

```
CREATE TABLE Course
(   CourseNo     CHAR (6),
    CrsDesc      VARCHAR (250),
    CrsUnits     SMALLINT,
CONSTRAINT  PKCourse PRIMARY KEY (CourseNo),
CONSTRAINT  UniqueCrsDesc UNIQUE (CrsDesc)    )
```

External FK Constraint Placement

```
CREATE TABLE Enrollment
(  OfferNo      INTEGER,
   StdNo        CHAR (11),
   EnrGrade     DECIMAL (3,2),
CONSTRAINT  PKEnrollment PRIMARY KEY
    (OfferNo, StdNo),
CONSTRAINT  FKOfferNo FOREIGN KEY (OfferNo)
    REFERENCES Offering,
CONSTRAINT  FKStdNo FOREIGN KEY (StdNo)
    REFERENCES Student )
```

Inline Constraint Placement

```
CREATE TABLE Offering
( OfferNo    INTEGER,
  CourseNo   CHAR (6) CONSTRAINT OffCourseNoReq NOT NULL,
  OffLocation VARCHAR (50),
  OffDays    CHAR (6),
  OffTerm    CHAR (6) CONSTRAINT OffTermReq NOT NULL,
  OffYear    INTEGER CONSTRAINT OffYearReq NOT NULL,
  FacNo      CHAR (11),
  OffTime    DATE,
CONSTRAINT PKOffering PRIMARY KEY (OfferNo),
CONSTRAINT FKCourseNo FOREIGN KEY (CourseNo)
    REFERENCES Course,
CONSTRAINT FKFacNo FOREIGN KEY (FacNo)
    REFERENCES Faculty )
```

Check Constraint Examples

```
CONSTRAINT ValidGPA CHECK ( StdGPA BETWEEN 0 AND 4 )

CONSTRAINT ValidStdClass
    CHECK ( StdClass IN ('FR', 'SO', 'JR', 'SR' )

CONSTRAINT OffYearValid CHECK (OffYear > 1970)

CONSTRAINT EnrollDropValid
    CHECK ( EnrolDate < DropDate )
```

(Structured Query Language)

A computer language containing statements for database definition, control and manipulation.

Major SQL Statements

Statement	Statement Type
CREATE TABLE	Definitional, Control
CREATE VIEW	Definitional
CREATE TYPE	Definitional
SELECT	Manipulation
INSERT, UPDATE, DELETE	Manipulation
COMMIT, ROLLBACK	Manipulation
CREATE TRIGGER	Control, Manipulation
GRANT, REVOKE	Control

Definition Statements
(Create Objects)

Manipulation Statements
(Retrieve or Change Rows)

Control Statements
(Ensure Proper Usage of a Database)

SELECT <list of column expressions>

SELECT <column name>

SELECT Statement Syntax

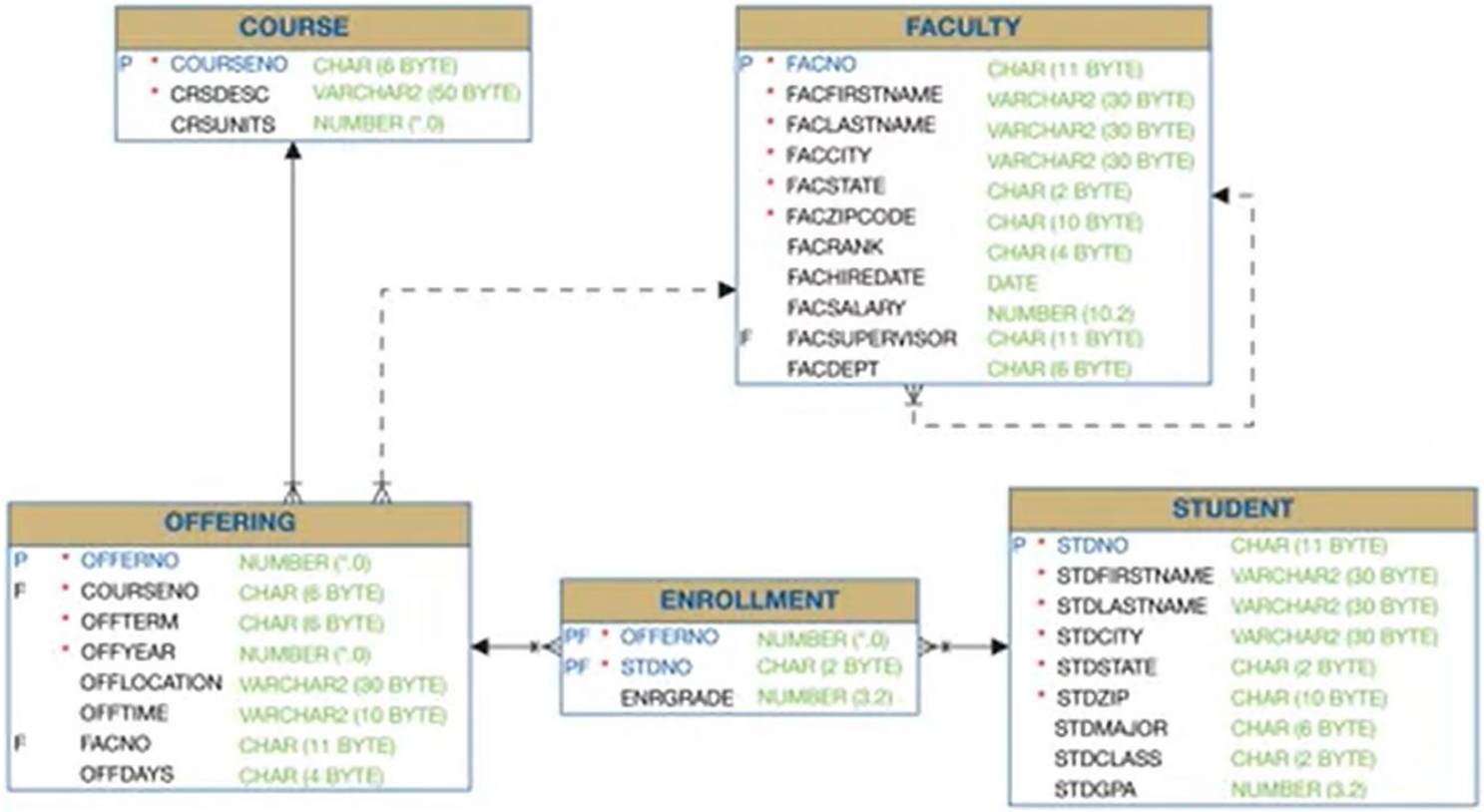
SELECT <list of column expressions>

FROM <list of tables and join operations>

WHERE <list of logical expressions for rows>

ORDER BY <list of sorting specifications>

University Database



SQL Data Type Examples

NUMERIC

integer, fixed decimal, floating point

STRING

CHAR OR VARCHAR

SELECT Statement: Example 1

```
SELECT * FROM Faculty;
```

	facno [PK] character (11)	facfirstname character varying (30)	faclastname character varying (30)	faccity character varying (30)	facstate character (2)	faczipcode character (10)	facrank character (4)	f d
1	543-21-0987	VICTORIA	EMMANUEL	BOTHELL	WA	98011-2242	PROF	2
2	765-43-2109	NICKI	MACON	BELLEVUE	WA	98015-9945	PROF	2
3	654-32-1098	LEONARD	FIBON	SEATTLE	WA	98121-0094	ASSC	2
4	098-76-5432	LEONARD	VINCE	SEATTLE	WA	98111-9921	ASST	2
5	876-54-3210	CRISTOPHER	COLAN	SEATTLE	WA	98114-1332	ASST	2
6	987-65-4321	JULIA	MILLS	SEATTLE	WA	98114-9954	ASSC	2

SELECT Statement: Example 2

```
SELECT *
 FROM Faculty
 WHERE FacNo = '543-21-0987';
```

	facno [PK] character (11)	facfirstname character varying (30)	faclastname character varying (30)	faccity character varying (30)	facstate character (2)	faczipcode character (10)	facrank character (4)
1	543-21-0987	VICTORIA	EMMANUEL	BOTHELL	WA	98011-2242	PROF

SELECT Statement: Example 3

```
SELECT FacFirstName, FacLastName, FacSalary
 FROM Faculty
 WHERE FacSalary > 65000 AND FacRank = 'PROF';
```

	facfirstname character varying (30)	faclastname character varying (30)	facsalary numeric (10,2)
1	VICTORIA	EMMANUEL	120000.00

SELECT Statement: Example 4

```
SELECT FacCity, FacState
 FROM Faculty
```

	faccity character varying (30)	facstate character (2)
1	BOTHELL	WA
2	BELLEVUE	WA
3	SEATTLE	WA
4	SEATTLE	WA
5	SEATTLE	WA
6	SEATTLE	WA

```
SELECT DISTINCT FacCity, FacState
 FROM Faculty;
```

	faccity character varying (30)	facstate character (2)
1	SEATTLE	WA
2	BOTHELL	WA
3	BELLEVUE	WA

SELECT Statement: Example 5

```
SELECT FacFirstName, FacLastName, FacCity,
       FacSalary*1.1 AS IncreasedSalary,
       FacHireDate
FROM Faculty
WHERE to_number(to_char(FacHireDate, 'YYYY')) > 2005;
```

	facfirstname character varying (30)	faclastname character varying (30)	faccity character varying (30)	increasedsalary numeric	fachiredate date
1	NICKI	MACON	BELLEVUE	71500.000	2006-04-11
2	CRISTOPHER	COLAN	SEATTLE	44000.000	2008-03-01
3	JULIA	MILLS	SEATTLE	82500.000	2009-03-15

Inexact Text Matching

```
SELECT *
 FROM Offering
 WHERE CourseNo LIKE 'IS%';
```

	offerno [PK] integer	courseno character (6)	offterm character (6)	offyear integer	offlocation character varying (3
1	1111	IS320	SUMMER	2017	BLM302
2	1234	IS320	FALL	2016	BLM302
3	2222	IS460	SUMMER	2016	BLM412
4	3333	IS320	SPRING	2017	BLM214
5	4321	IS320	FALL	2016	BLM214
6	4444	IS320	WINTER	2017	BLM302
7	5678	IS480	WINTER	2017	BLM302
8	5679	IS480	SPRING	2017	BLM412
9	8888	IS320	SUMMER	2017	BLM405
10	9876	IS460	SPRING	2017	BLM307

Inexact Text Matching

```
SELECT *
 FROM Offering
```

	offerno [PK] integer	courseno character (6)	offterm character (6)	offyear integer	offlocation character varying (30)
1	1111	IS320	SUMMER	2017	BLM302
2	1234	IS320	FALL	2016	BLM302
3	2222	IS460	SUMMER	2016	BLM412
4	3333	IS320	SPRING	2017	BLM214
5	4321	IS320	FALL	2016	BLM214
6	4444	IS320	WINTER	2017	BLM302
7	5555	FIN300	WINTER	2016	BLM207
8	5678	IS480	WINTER	2017	BLM302
9	5679	IS480	SPRING	2017	BLM412
10	6666	FIN450	WINTER	2017	BLM212
11	7777	FIN480	SPRING	2017	BLM305
12	8888	IS320	SUMMER	2017	BLM405
13	9876	IS460	SPRING	2017	BLM307

Example of SQL-Variant SELECT Statements

```
SELECT FacFirstName, FacLastName, FacHireDate
 FROM Faculty
 WHERE FacHireDate BETWEEN '1-Jan-2008'
   AND '31-Dec-2009';
```

ORACLE


```
SELECT FacFirstName, FacLastName, FacHireDate
 FROM Faculty
 WHERE FacHireDate BETWEEN '2008-01-01'
   AND '2009-12-31';
```

Testing For Null Values

```
SELECT OfferNo, CourseNo, FacNo, OffTerm, OffYear
 FROM Offering
 WHERE FacNo IS NULL AND OffTerm = 'SUMMER'
   AND OffYear = 2017;
```

	offerno [PK] integer	courseno character (6)	facno character (11)	offterm character (6)	offyear integer
1	1111	IS320	[null]	SUMMER	2017

Mixing AND and OR

```
SELECT OfferNo, CourseNo, FacNo, OffTerm, OffYear
 FROM Offering
 WHERE (OffTerm = 'FALL' AND OffYear = 2016)
   OR (OffTerm = 'WINTER' AND OffYear = 2017);
```

	offerno [PK] integer	courseno character (6)	facno character (11)	offterm character (6)	offyear integer
1	1234	IS320	098-76-5432	FALL	2016
2	4321	IS320	098-76-5432	FALL	2016
3	4444	IS320	543-21-0987	WINTER	2017
4	5678	IS480	987-65-4321	WINTER	2017
5	6666	FIN450	987-65-4321	WINTER	2017

Natural JOIN

Offering

OfferNo	FacNo
1111	111-11-1111
2222	222-22-2222
3333	111-11-1111

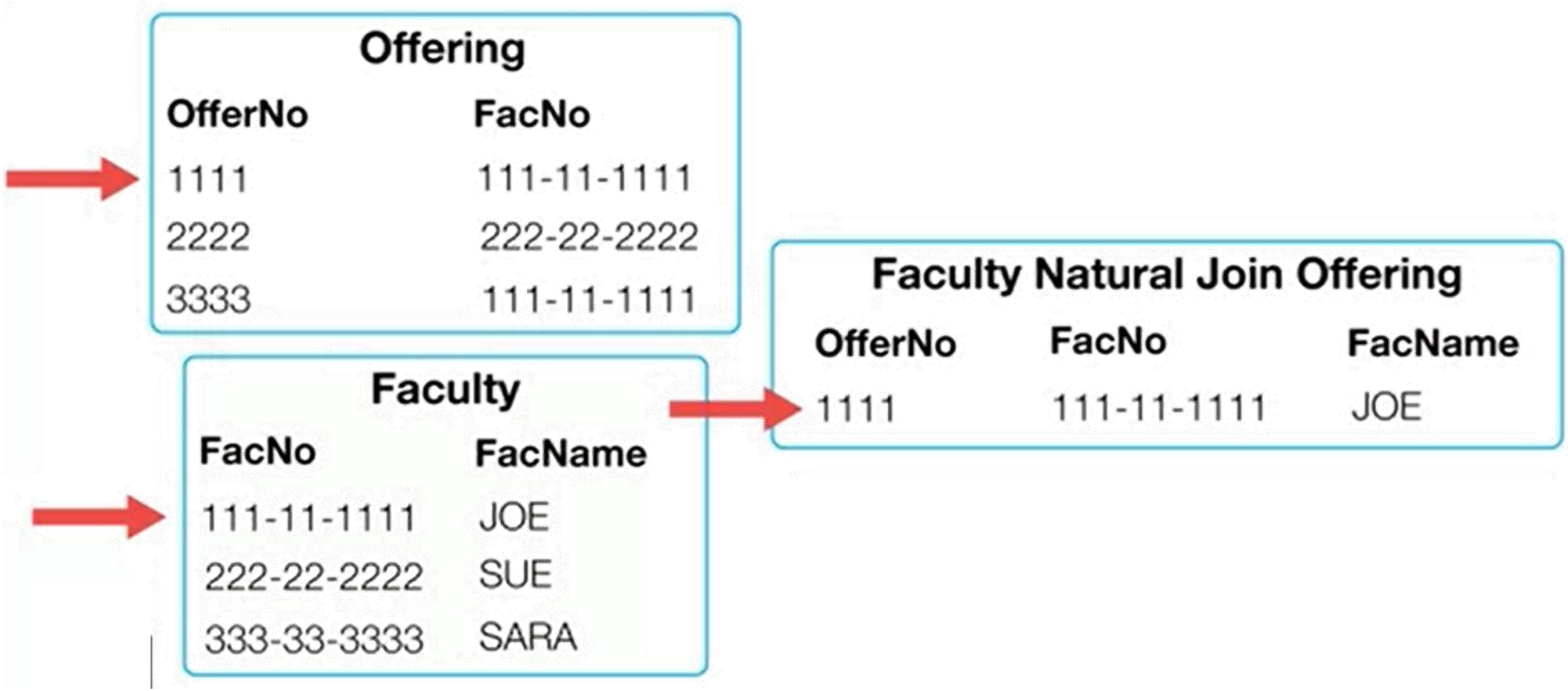
Faculty

FacNo	FacName
111-11-1111	JOE
222-22-2222	SUE
333-33-3333	SARA

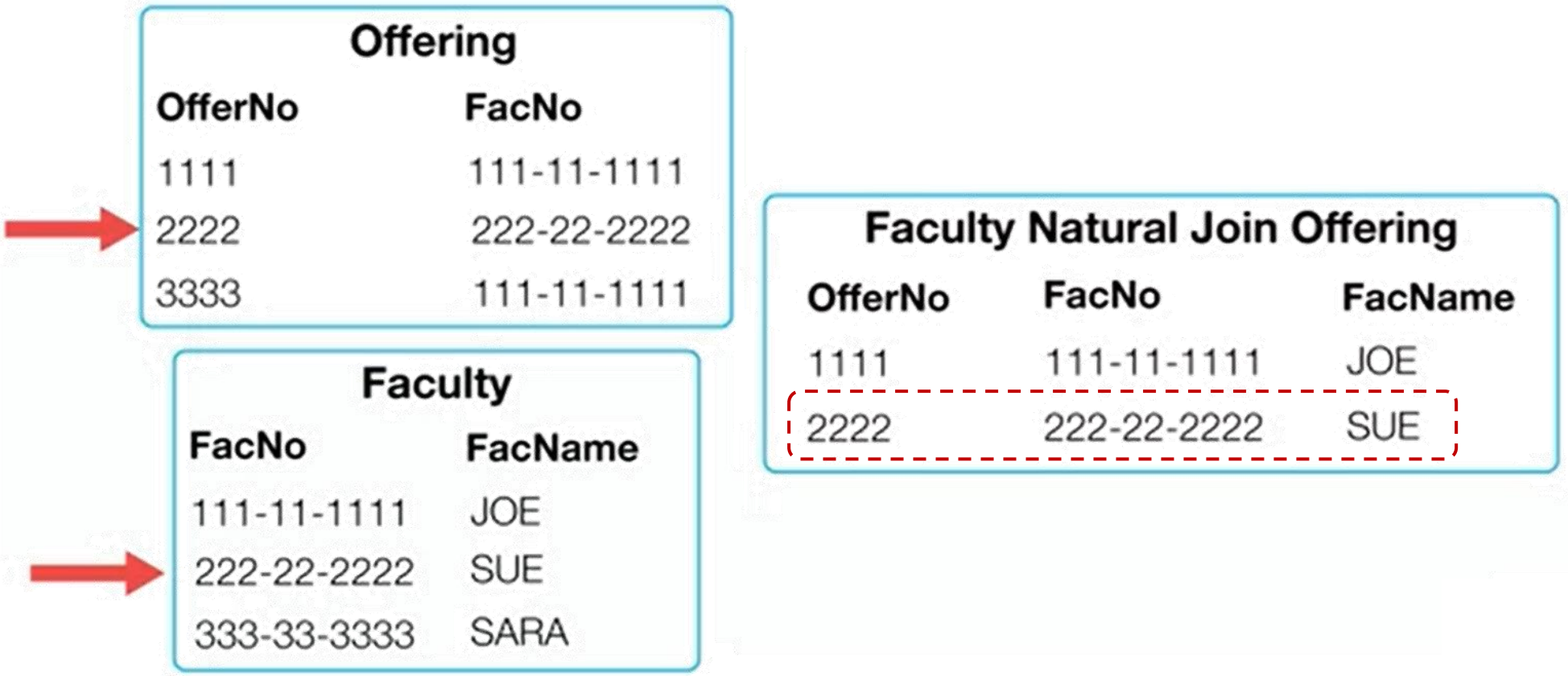
Unqualified

When the column name is alone without the table identifier.

Natural JOIN: Example



Natural JOIN: Example



Natural JOIN: Example

Offering

OfferNo	FacNo
1111	111-11-1111
2222	222-22-2222
3333	111-11-1111

Faculty

FacNo	FacName
111-11-1111	JOE
222-22-2222	SUE
333-33-3333	SARA

Faculty Natural Join Offering

OfferNo	FacNo	FacName
1111	111-11-1111	JOE
2222	222-22-2222	SUE
3333	111-11-1111	JOE

Using JOIN Operations To Support Business Decisions

JOIN Operator

The JOIN operator builds a new table by combining rows from two tables that match on a join condition.

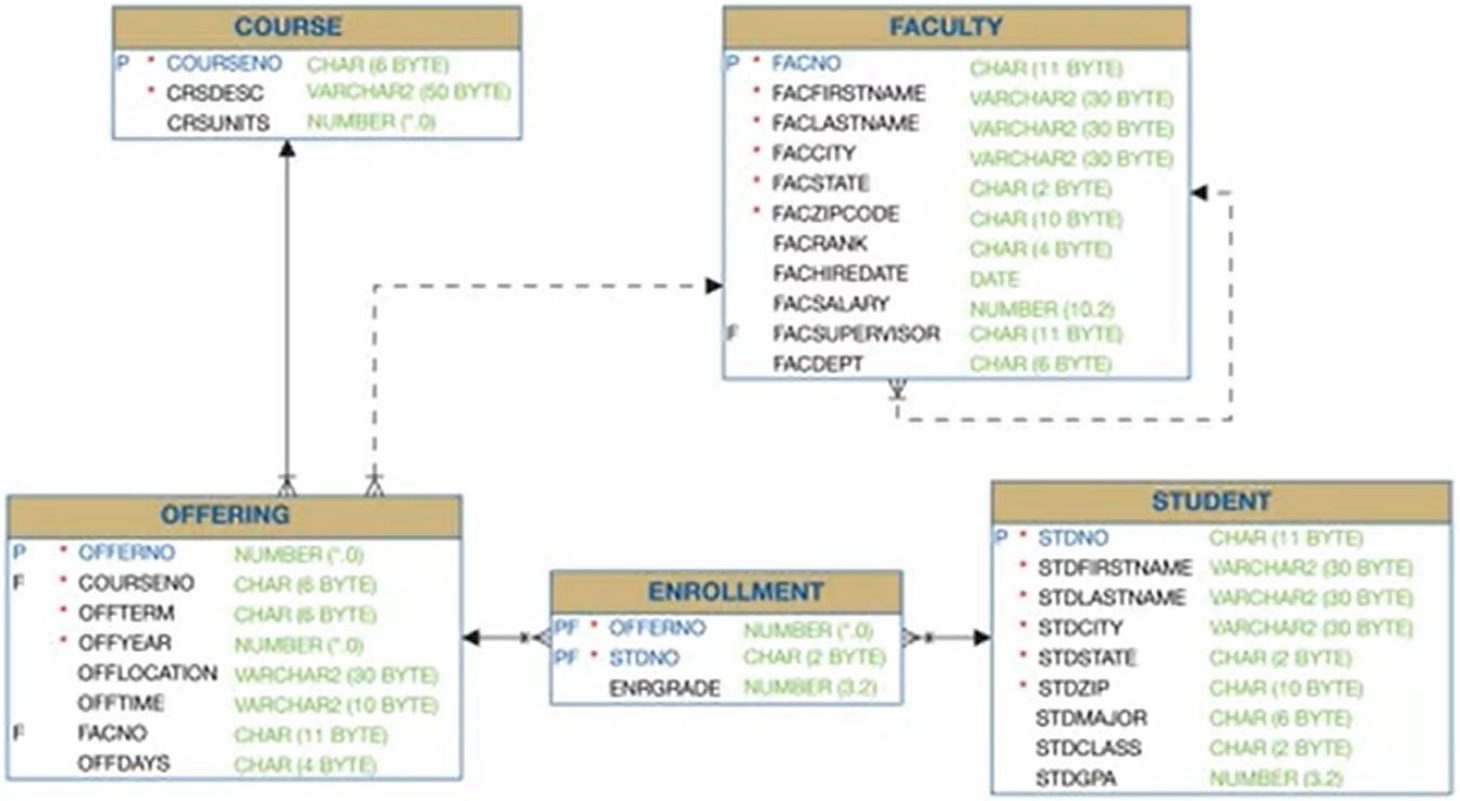
(Equality Join)

When the join condition involves equality.

Most Common JOIN Condition

- Combining tables using a condition in which the PK of a parent row equals the FK of a child table is the most common join condition.
- The natural join operator is the most common join operation.
- The join condition is equality (EQUI-JOIN), one of the join columns is removed and the join columns have the same unqualified name.
- Typically, the join columns are a combination of PK and FK.

University Database



Cross Product Style

```
SELECT OfferNo, CourseNo, FacFirstName, FacLastName
 FROM Offering, Faculty
 WHERE OffTerm = 'FALL' AND OffYear = 2016
      AND FacRank = 'ASST' AND CourseNo LIKE 'IS%'
      AND Faculty.FacNo = Offering.FacNo;
```

	offerno integer	courseno character (6)	facfirstname character varying (30)	faclastname character varying (30)
1	1234	IS320	LEONARD	VINCE
2	4321	IS320	LEONARD	VINCE

JOIN Operator Style

```
SELECT OfferNo, CourseNo, FacFirstName, FacLastName
  FROM Offering, Faculty
     ON Faculty.FacNo = Offering.FacNo
  WHERE OffTerm = 'FALL' AND OffYear = 2016
        AND FacRank = 'ASST' AND CourseNo LIKE 'IS%';
```

	offerno integer	courseno character (6)	facfirstname character varying (30)	faclastname character varying (30)
1	1234	IS320	LEONARD	VINCE
2	4321	IS320	LEONARD	VINCE

Name Qualification

Column names used in a statement must be qualified with its associated table name if the column name alone is ambiguous.

Ambiguity occurs if the same unqualified column name occurs in more than one table in the statement.

Name Qualification Example

```
SELECT OfferNo, CourseNo, FacFirstName, FacLastName
  FROM Offering INNER JOIN Faculty
    ON Faculty.FacNo = Offering.FacNo
 WHERE OffTerm = 'FALL' AND OffYear = 2016
    AND FacRank = 'ASST' AND CourseNo LIKE 'IS%';
```

	facno [PK] character (11)	facfirstname character varying (30)	faclastname character varying (30)	faccity character varying (30)
1	543-21-0987	VICTORIA	EMMANUEL	BOTHELL
2	765-43-2109	NICKI	MACON	BELLEVUE
3	654-32-1098	LEONARD	FIBON	SEATTLE
4	098-76-5432	LEONARD	VINCE	SEATTLE
5	876-54-3210	CRISTOPHER	COLAN	SEATTLE
6	987-65-4321	JULIA	MILLS	SEATTLE

	offerno [PK] integer	courseno character (6)	offterm character (6)	offyear integer	offlocation character varying (30)	offtime character varying (10)	facno character (11)	offdays character (4)
1	1111	IS320	SUMMER	2017	BLM302	10:30:00	[null]	MW
2	1234	IS320	FALL	2016	BLM302	10:30:00	098-76-5432	MW
3	2222	IS460	SUMMER	2016	BLM412	13:30:00	[null]	TTH
4	3333	IS320	SPRING	2017	BLM214	08:30:00	098-76-5432	MW
5	4321	IS320	FALL	2016	BLM214	15:30:00	098-76-5432	TTH
6	4444	IS320	WINTER	2017	BLM302	15:30:00	543-21-0987	TTH
7	5555	FIN300	WINTER	2016	BLM207	08:30:00	765-43-2109	MW
8	5678	IS480	WINTER	2017	BLM302	10:30:00	987-65-4321	MW
9	5679	IS480	SPRING	2017	BLM412	15:30:00	876-54-3210	TTH
10	6666	FIN450	WINTER	2017	BLM212	10:30:00	987-65-4321	TTH
11	7777	FIN480	SPRING	2017	BLM305	13:30:00	765-43-2109	MW
12	8888	IS320	SUMMER	2017	BLM405	13:30:00	654-32-1098	MW
13	9876	IS460	SPRING	2017	BLM307	13:30:00	654-32-1098	TTH

Cross Product Style With 3 Tables

```
SELECT  OfferNo, Offering.CourseNo, OffDays,
        CrsUnits, OffLocation, OffTime
FROM Faculty, Course, Offering
WHERE Faculty.FacNo = Offering.FacNo
    AND Offering.CourseNo = Course.CourseNo
    AND OffYear = 2016 AND OffTerm = 'FALL'
    AND FacFirstName = 'LEONARD'
    AND FacLastName = 'VINCE';
```

JOIN Operator Style With 3 Tables

```
SELECT OfferNo, Offering.CourseNo, OffDays,
            CrsUnits, OffLocation, OffTime
 FROM Offering INNER JOIN Course
   ON Offering.CourseNo = Course.CourseNo
   INNER JOIN Faculty ON Offering.FacNo = Faculty.FacNo
 WHERE OffYear = 2016 AND OffTerm = 'FALL'
     AND FacFirstName = 'LEONARD'
     AND FacLastName = 'VINCE';
```

Table Statistics

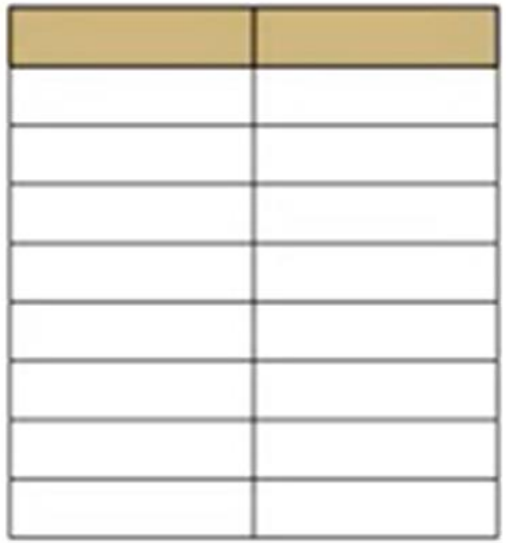


Table Statistics

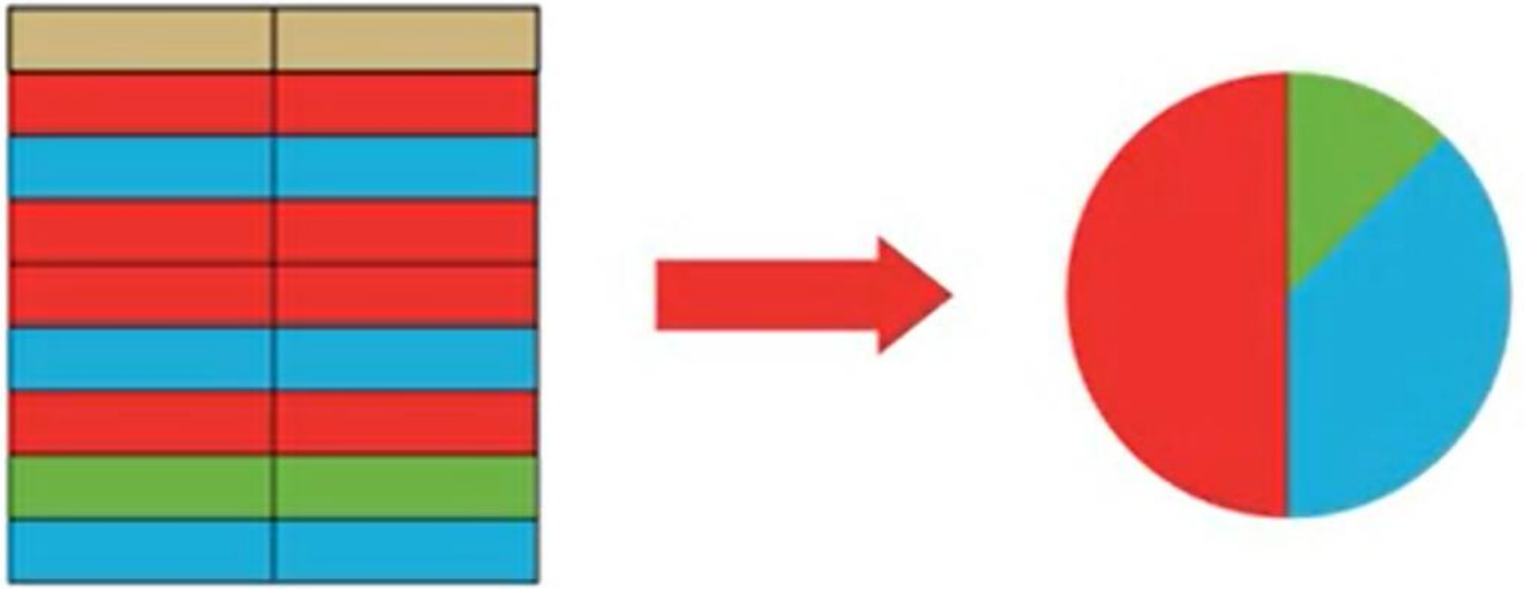


Table Statistics: Example

StdNo	StdMajor	StdGPA
123-45-6789	ACCT	3.00
234-56-7890	FIN	2.70
345-67-8901	ACCT	3.50
456-78-9012	FIN	2.50
567-89-0123	ACCT	2.00
678-90-1234	ACCT	4.00

Average ACCT GPA

3.125

Row Summaries

- Important for decision-making tasks
- Row summary details
  - Result contains statistical (aggregate) functions
  - Conditions involve statistical functions
- SQL keywords
  - Aggregate functions in the result list such as AVG and SUM
  - GROUP BY: summary columns
  - HAVING: summary conditions

Individual Rows: Sort By Faculty Rank

```
SELECT FacNo, FacRank, FacSalary
FROM Faculty
ORDER BY FacRank;
```

FACNO	FACRANK	FACSALARY
654-32-1098	ASSC	70000
987-65-4321	ASSC	75000
876-54-3210	ASST	40000
098-76-5432	ASST	35000
765-43-2109	PROF	65000
543-21-0987	PROF	120000

Row Summaries: Average Salary For Each Faculty Rank

```
SELECT FacRank,
       AVG (FacSalary) AS AvgSalary
  FROM Faculty
  GROUP BY FacRank
ORDER BY FacRank;
```

FACRANK	AVGSALARY
ASSC	72500
ASST	37500
PROF	92500

Filtering Rows and Groups

```
SELECT StdMajor, AVG (StdGPA) AS AvgGpa
FROM Student
WHERE StdClass IN ('JR', 'SR')
GROUP BY StdMajor;
```

STDMAJOR	AVGGPA
ACCT	3.5
FIN	2.8
IS	3.15

Filtering Rows and Groups

```
SELECT StdMajor, AVG (StdGPA) AS AvgGpa
FROM Student
WHERE StdClass IN ('JR', 'SR')
GROUP BY StdMajor
HAVING AVG (StdGPA) > 3.1;
```

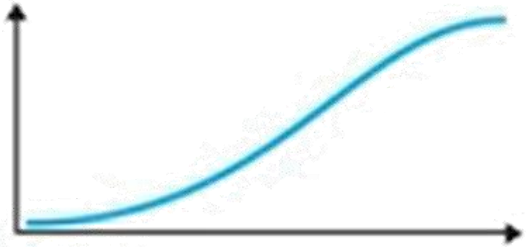
STDMAJOR	AVGGPA
ACCT	3.5
IS	3.15

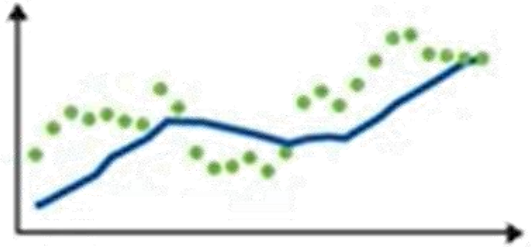
Query Clause Evaluation Order

Evaluation Order Summaries

- Row operations before group operations
  - FROM and WHERE before GROUP BY and HAVING
  - Check row operations first
- Grouping occurs only one time
- Use small sample tables

Analytical Calculations





Summary: What We Have Learned

- Transactions are units of information work that must be processed together. DBMS provide services that ensure reliable transaction processing with no data losses from concurrent users and failures after completion.

- Primary (PK) and Foreign (FK) Keys are used to enforce constraints when data are inserted into the database. They ensure data integrity between the related tables.

- SQL is non-procedural access and a crucial DBMS feature. SELECT statement is important and complex and need lots of practice to master the query formation.

- JOIN is an essential operator in query formulation. It is used to combine data from different tables to create wider cross-functional aggregated insights beyond the scope of a single table can offer.

- Summarisation queries are common for business intelligence. GROUP BY clause is used to calculate summary data for decision making.

Q&A