CS 310: Database Systems

Tung Nguyen, PhD

Department of Computer Science and Engineering

Introduction

- Instructor
  - Tung Nguyen
    - Email: tung@tamu.edu
    - Office: PETR 114. Office hour: 2-3 pm Tuesday or online meeting by request
    - Experience: AI for Software Engineering
- Teaching Assistant

More introduction

- 2000 – 2004: BS in Computer Science at Hanoi National University of Education (Vietnam)
- 2004 – 2007: Lecturer at Hanoi National University of Education
- 2007 – 2013: PhD in Computer Engineering at Iowa State University
- 2013 – 2017: Assistant Professor at Utah State University
- 2017 – 2023: Assistant Professor at Auburn University
- 2023 – now: Associate Professor (instructional) at Texas A & M University
  - Teaching Database Systems (310), Senior Design (482)

Teaching

- Experience
  - 10+ years as a professor in US
- Topics: programming, data structures, algorithms, software engineering, data science, machine learning, artificial intelligence
- Languages: Java, C/C++, R, Python, SQL

Research

- Intelligent tools for software engineering, esp. write code and fix bugs
  - Code completion, exception handling, bug detection, clone detection...
- Advanced algorithms and tools to discover knowledge and patterns in software engineering
  - Based on data mining, machine learning, natural language processing...
- Advanced program analysis techniques and models for software code
  - Source code, bytecode
- Advanced text analysis techniques for software code and documents
  - Comments, identifiers, bug reports, user reviews...
- See more at Google Scholar: https://scholar.google.com/citations?user=bpqJ-N0AAAAJ

Grading

Homework and assignments: 30%

- Around 10 in total (weekly or bi-weekly)

Course project: 35%

- Developing a software system using databases

Final exam: 20%

- Oral exam for Honors students, written for others
- Have several questions about course project

Attendance, activities and quizzes: 15%

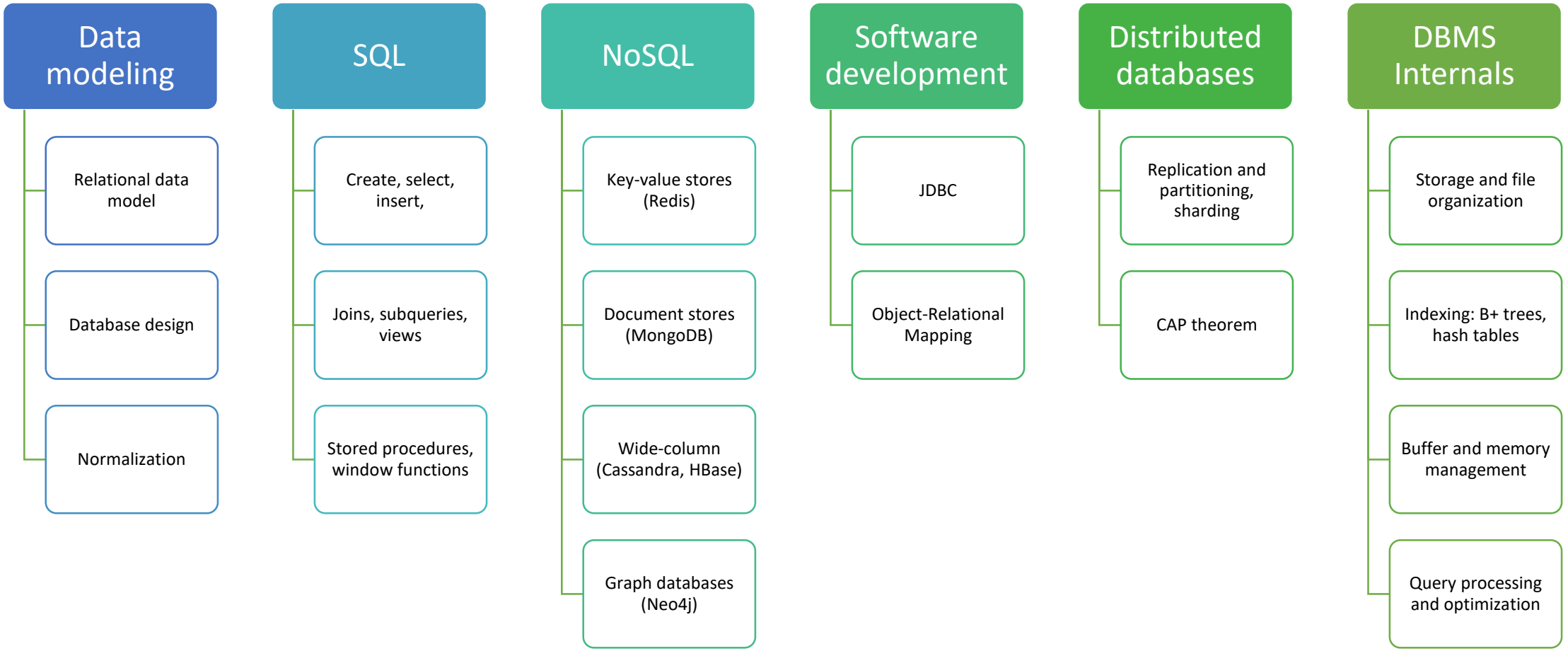
- Attendance is enforced by
  - Not providing lecture slides (several are reused from famous sources)
  - Daily quizzes (students need to work directly in class)

Topics

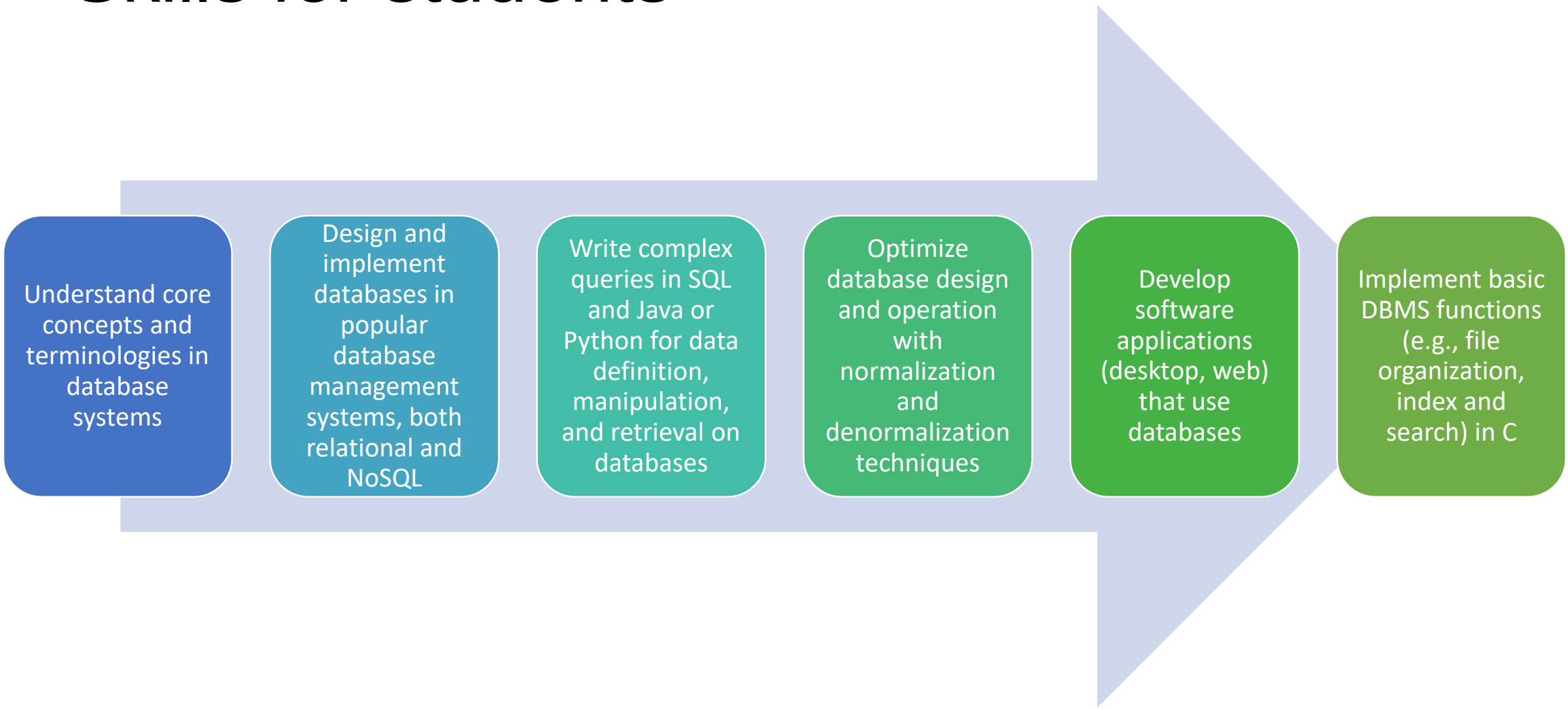
Introduction to database systems and their applications

- Fundamental concepts
- Design principles, and
- Practical use cases of
- Databases and
- Database management systems (DBMS)

Topics


Data modeling
Relational data model
Database design
Normalization
SQL
Create, select, insert,
Joins, subqueries, views
Stored procedures, window functions
NoSQL
Key-value stores (Redis)
Document stores (MongoDB)
Wide-column (Cassandra, HBase)
Graph databases (Neo4j)
Software development
JDBC
Object-Relational Mapping
Distributed databases
Replication and partitioning, sharding
CAP theorem
DBMS Internals
Storage and file organization
Indexing: B+ trees, hash tables
Buffer and memory management
Query processing and optimization

Skills for students



More information

- This course is programming-intensive
  - Design, coding, testing, reading, writing
- Policies
  - Students can use any helpful sources, including AI
    - External sources should be cited and acknowledged clearly

TABLE 5.1 A SAMPLE REPORT LAYOUT

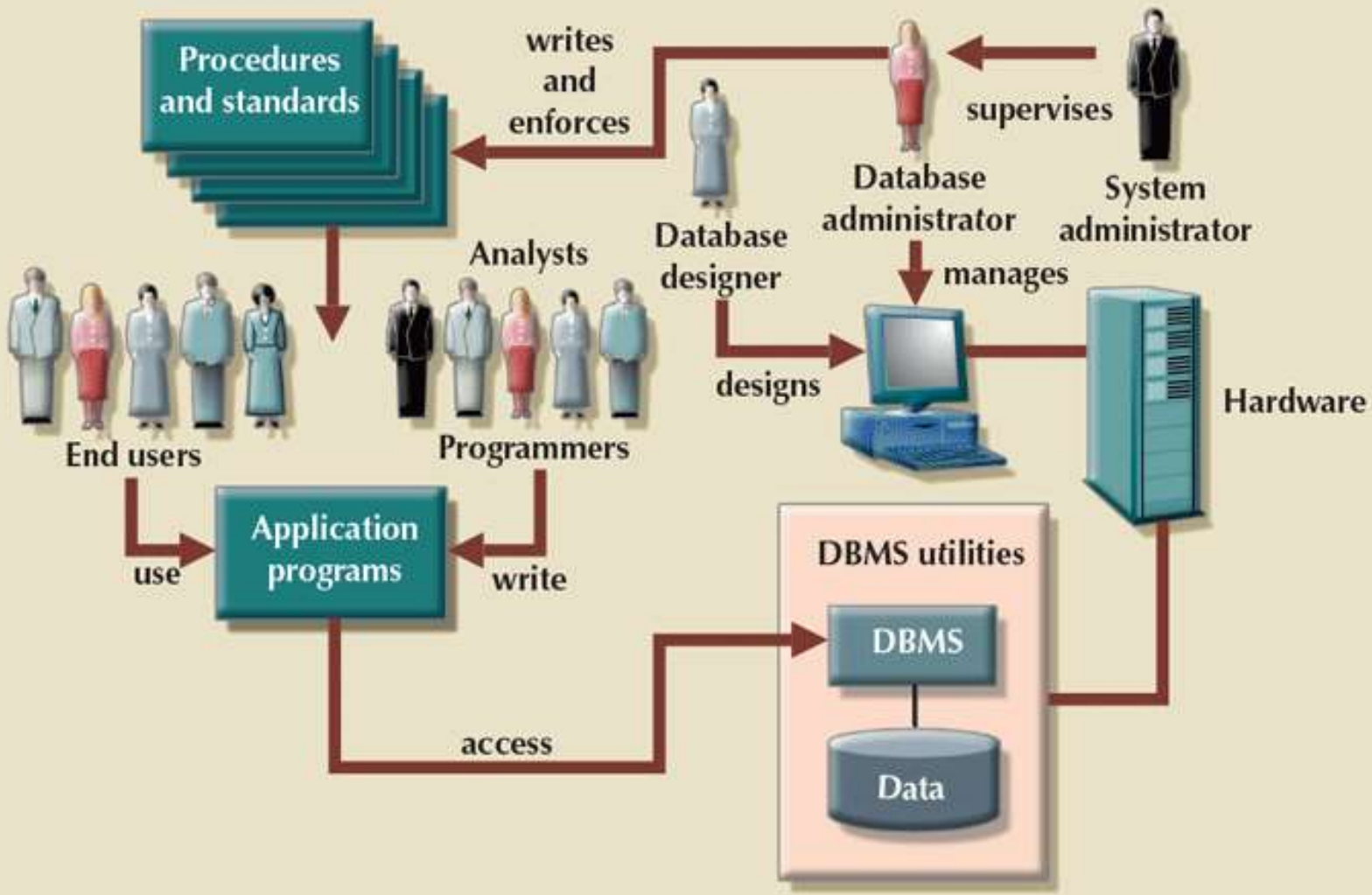
PROJ. NUM.	PROJECT NAME	EMPLOYEE NUMBER	EMPLOYEE NAME	JOB CLASS.	CHG/HOUR	HOURS BILLED	TOTAL CHARGE
15	Evergreen	103	June E. Arbough	Elec. Engineer	$84.50	23.8	$2,011.10
		101	John G. News	Database Designer	$105.00	19.4	$2,037.00
		105	Alice K. Johnson *	Database Designer	$105.00	35.7	$3,748.50
		106	William Smithfield	Programmer	$35.75	12.6	$450.45
		102	David H. Senior	Systems Analyst	$96.75	23.8	$2,302.65
				Subtotal			**$10,549.70**
18	Amber Wave	114	Annelise Jones	Applications Designer	$48.10	24.6	$1,183.26
		118	James J. Frommer	General Support	$18.36	45.3	$831.71
		104	Anne K. Ramoras *	Systems Analyst	$96.75	32.4	$3,135.70
		112	Darlene M. Smithson	DSS Analyst	$45.95	44.0	$2,021.80
				Subtotal			**$7,171.47**
22	Rolling Tide	105	Alice K. Johnson	Database Designer	$105.00	64.7	$6,793.50
		104	Anne K. Ramoras	Systems Analyst	$96.75	48.4	$4,682.70
		113	Delbert K. Joenbrood *	Applications Designer	$48.10	23.6	$1,135.16
		111	Geoff B. Wabash	Clerical Support	$26.87	22.0	$591.14
		106	William Smithfield	Programmer	$35.75	12.8	$457.60
				Subtotal			**$13,660.10**
25	Starflight	107	Maria D. Alonzo	Programmer	$35.75	24.6	$879.45
		115	Travis B. Bawangi	Systems Analyst	$96.75	45.8	$4,431.15
		101	John G. News *	Database Designer	$105.00	56.3	$5,911.50
		114	Annelise Jones	Applications Designer	$48.10	33.1	$1,592.11
		108	Ralph B. Washington	Systems Analyst	$96.75	23.6	$2,283.30
		118	James J. Frommer	General Support	$18.36	30.5	$559.98
		112	Darlene M. Smithson	DSS Analyst	$45.95	41.4	$1,902.33
				Subtotal			**$17,559.82**
				Total			**$48,941.09**

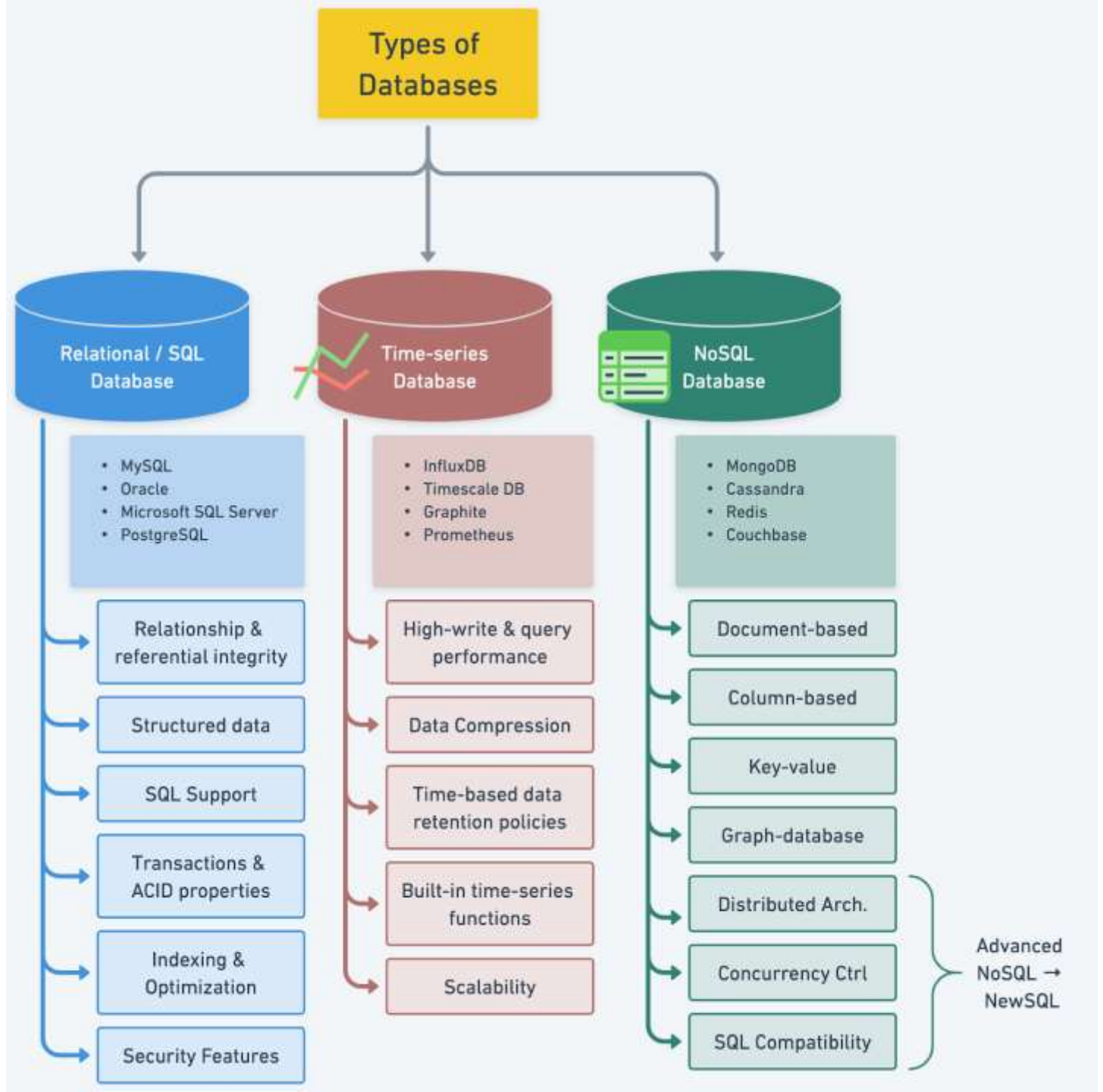
* Indicates project leader

The Database System Environment

- Database system: organization of components that define and regulate the collection, storage, management, and use of data within a database environment
  - Hardware
  - Software
  - People
  - Procedures
  - Data

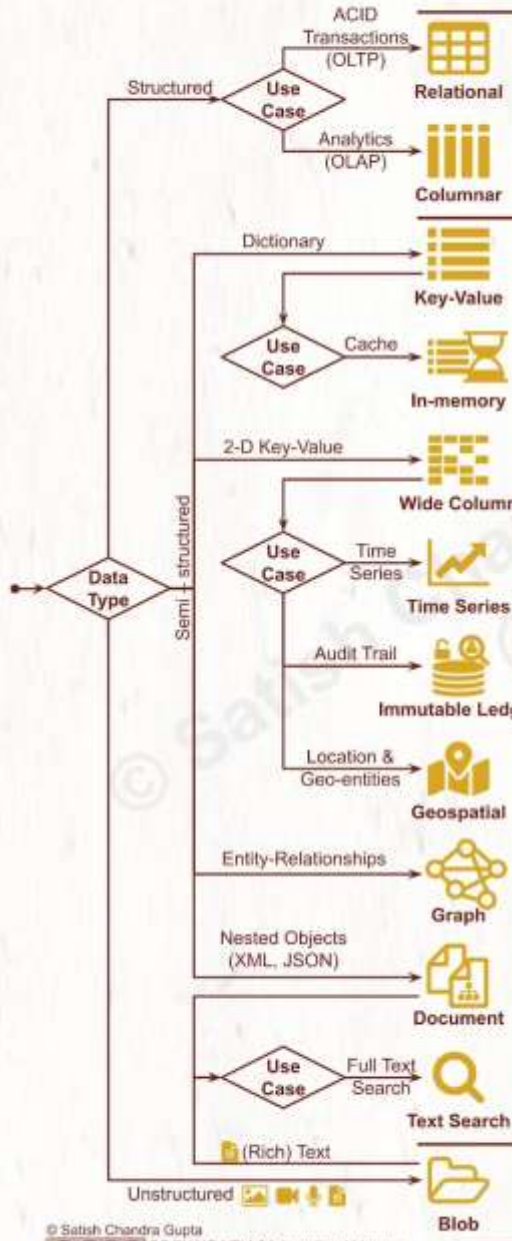
FIGURE 1.11 THE DATABASE SYSTEM ENVIRONMENT




Types of Databases
Relational / SQL Database
Time-series Database
NoSQL Database
MySQL
Oracle
Microsoft SQL Server
PostgreSQL
InfluxDB
Timescale DB
Graphite
Prometheus
MongoDB
Cassandra
Redis
Couchbase
Relationship & referential integrity
Structured data
SQL Support
Transactions & ACID properties
Indexing & Optimization
Security Features
High-write & query performance
Data Compression
Time-based data retention policies
Built-in time-series functions
Scalability
Document-based
Column-based
Key-value
Graph-database
Distributed Arch.
Concurrency Ctrl
SQL Compatibility
Advanced NoSQL → NewSQL

SQL vs. NoSQL: Cheatsheet for AWS, Azure, and Google Cloud

scgupta.link/datastores



	aws	Azure	Google Cloud	Cloud Agnostic
Relational	RDS, Aurora	Azure SQL Database	Cloud SQL, Cloud Spanner	SQL Server, Oracle, DB2, MySQL, PostgreSQL
Columnar	RedShift	Azure Synapse	BigQuery	Snowflake, ClickHouse, Druid, Pinot, Databricks
Key-Value	DynamoDB	Cosmos DB	BigTable	Redis, ScyllaDB, Ignite
In-memory	ElastiCache	Azure Cache for Redis	Memory-store	Redis, Memcached, Hazelcast, Ignite
Wide Column	Keyspaces	Cosmos DB	BigTable	HBase, Cassandra, ScyllaDB
Time Series	Timestream	Cosmos DB	BigTable, BigQuery	OpenTSDB, InfluxDB, ScyllaDB
Immutable Ledger	Quantum Ledger Database (QLDB)	Azure SQL Database Ledger	✖	Hyperledger Fabric
Geospatial	Keyspaces	Cosmos DB	BigTable, BigQuery	Solr, PostGIS, MongoDB (GeoJSON)
Graph	Neptune	Cosmos DB	JanusGraph + BigTable	OrientDB, Neo4J, Giraph
Document	Document DB	Cosmos DB	Firestore	MongoDB, Couchbase, Solr
Text Search	Open-Search, Cloud-Search	Cognitive Search	Search APIs on Datastores	Elastic-Search, Solr, Elassandra
Blob	S3	Blob Storage	Cloud Storage	HDFS, MinIO

NoSQL

scgupta.me
twitter.com/scgupta
linkedin.com/in/scgupta

Data Modeling and Data Models

- Model: abstraction of a more complex real-world object or event
- Data model: simple representation of complex real-world data structures
  - Useful for supporting a specific problem domain
- Data modeling: creating a specific data model for a problem domain

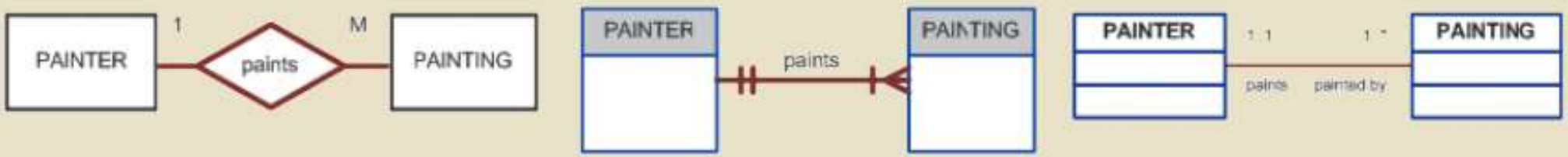
Data Model Basic Building Blocks

- Entity: person, place, thing, or event about which data will be collected and stored
  - Attribute: characteristic of an entity
  - Relationship: association among entities
    - One-to-many (1:M OR 1..*)
    - Many-to-many (M:N or *..*)
    - One-to-one (1:1 OR 1..1)
  - Constraint: restriction placed on data
    - Ensures data integrity

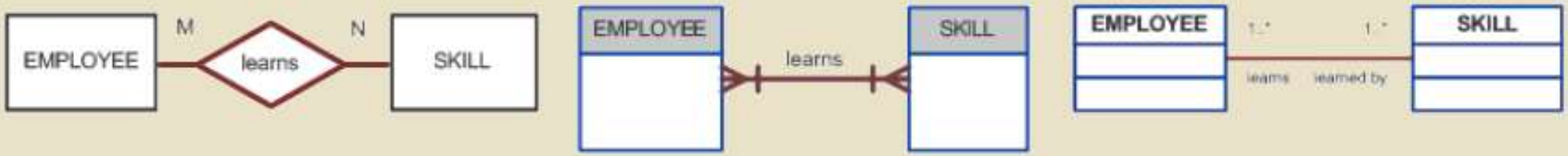
FIGURE 2.3 THE ER MODEL NOTATIONS

Chen Notation ***Crow's Foot Notation*** *UML Class Diagram Notation*

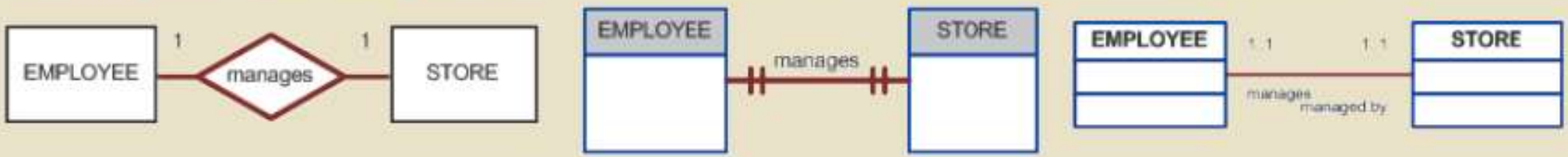
A One-to-Many (1:M) Relationship: a PAINTER can paint many PAINTINGs; each PAINTING is painted by one PAINTER.



A Many-to-Many (M:N) Relationship: an EMPLOYEE can learn many SKILLs; each SKILL can be learned by many EMPLOYEEs.



A One-to-One (1:1) Relationship: an EMPLOYEE manages one STORE; each STORE is managed by one EMPLOYEE.



Data Model Basic Building Blocks

- If an attribute has complex structure
  - Break into smaller attributes
    - Name = First Name + Last Name
  - Model as entity
    - Address = Street Number + Street + City + Zip…
- A many-to-many relationship could be represented as an entity
  - Customer <buy> Product → Order
  - Employee <work for> Project → Billing
  - Employee <attend> Class → Registration Record

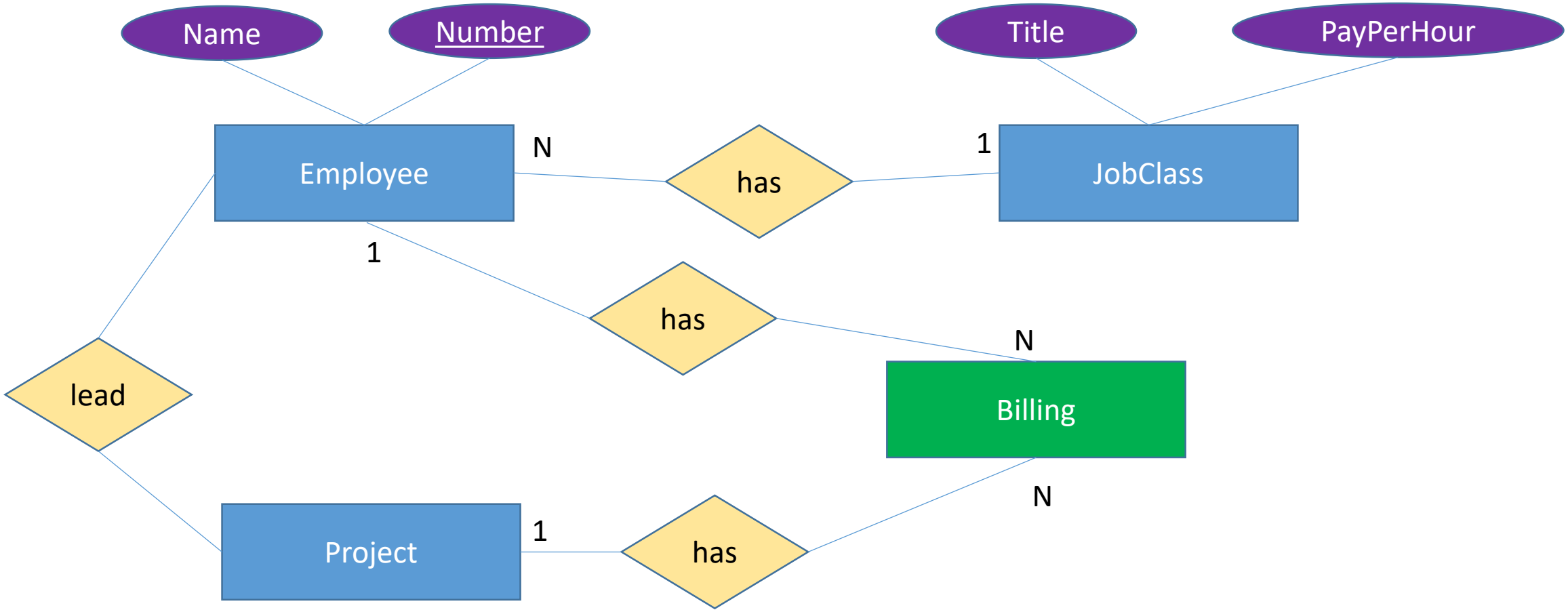
Data Model Basic Building Blocks

- Key
  - Attribute(s) uniquely identify or refer to an entity
  - Primary key: attributes identify a unique entity
    - Project number, Employee number, Job ID, Student UIN
  - Foreign key: attributes refer to another related entity
    - Job ID in Employee entity, Student UIN in Submission entity

Quiz 2: Analyze data in the project management sample

PROJ. NUM.	PROJECT NAME	EMPLOYEE NUMBER	EMPLOYEE NAME	JOB CLASS.	CHG/HOUR	HOURS BILLED	TOTAL CHARGE
15	Evergreen	103	June E. Arbough	Elec. Engineer	$84.50	23.8	$2,011.10
		101	John G. News	Database Designer	$105.00	19.4	$2,037.00
		105	Alice K. Johnson *	Database Designer	$105.00	35.7	$3,748.50
		106	William Smithfield	Programmer	$35.75	12.6	$450.45
		102	David H. Senior	Systems Analyst	$96.75	23.8	$2,302.65
				Subtotal			**$10,549.70**
18	Amber Wave	114	Annelise Jones	Applications Designer	$48.10	24.6	$1,183.26
		118	James J. Frommer	General Support	$18.36	45.3	$831.71
		104	Anne K. Ramoras *	Systems Analyst	$96.75	32.4	$3,135.70
		112	Darlene M. Smithson	DSS Analyst	$45.95	44.0	$2,021.80
				Subtotal			**$7,171.47**
22	Rolling Tide	105	Alice K. Johnson	Database Designer	$105.00	64.7	$6,793.50
		104	Anne K. Ramoras	Systems Analyst	$96.75	48.4	$4,682.70
		113	Delbert K. Joenbrood *	Applications Designer	$48.10	23.6	$1,135.16
		111	Geoff B. Wabash	Clerical Support	$26.87	22.0	$591.14
		106	William Smithfield	Programmer	$35.75	12.8	$457.60
				Subtotal			**$13,660.10**
25	Starflight	107	Maria D. Alonzo	Programmer	$35.75	24.6	$879.45
		115	Travis B. Bawangi	Systems Analyst	$96.75	45.8	$4,431.15
		101	John G. News *	Database Designer	$105.00	56.3	$5,911.50
		114	Annelise Jones	Applications Designer	$48.10	33.1	$1,592.11
		108	Ralph B. Washington	Systems Analyst	$96.75	23.6	$2,283.30
		118	James J. Frommer	General Support	$18.36	30.5	$559.98
		112	Darlene M. Smithson	DSS Analyst	$45.95	41.4	$1,902.33
				Subtotal			**$17,559.82**
				Total			**$48,941.09**

* Indicates project leader


Name
Number
Title
PayPerHour
Employee
N
has
1
JobClass
1
has
N
lead
Billing
Project
1
has
N

Example of keys

Physician (ID, Name, …)

Patient (ID, Name, PhysID*, …)

Club (ID, Name, …)

Player (ID, Name, ?*, …)

Order (OrdID, Date, …, ?*)

Customer (ID, Name, …, ?*)

Dept (DeptID, Name, …, ?*)

Employee (EID, Name, …, ?*)

Course (CourseID, Name, …, ?*)

Class (ClassID, Meets, …, ?*)

Student (SID, Name, …, ?*)

Registration (?)

	A	B	C	D	E	F	G	H	I
1	Sales Representative	Location	Region	Customer	Order Date	Item	Quantity	Price	Total Sale Amount
2	Sara Snyder	New York	East	Phyllis Johnston	2016-10-30	Things	1	17.83	17.83
3	Sara Snyder	New York	East	Kimberly Little	2016-05-23	Junk	3	12.42	37.26
4	Frances Warren	Massachusetts	East	Justin Dixon	2016-09-27	Widgets	4	53.35	213.4
5	Sara Snyder	Massachusetts	East	Shirley Rivera	2016-02-12	Junk	5	12.42	62.1
6	Diane Gonzalez	Oregon	West	Marilyn Franklin	2016-02-14	Things	8	17.83	142.64
7	Patrick Graham	Washington	West	Henry Sanders	2016-04-11	Widgets	4	53.35	213.4
8	Sara Snyder	Connecticut	East	Benjamin Phillips	2016-09-02	Junk	4	12.42	49.68
9	Frances Warren	New Jersey	East	Theresa Torres	2016-11-26	Junk	4	12.42	49.68
10	Patrick Graham	Oregon	West	Roger Bell	2016-07-13	Junk	10	12.42	124.2
11	Sara Snyder	New Jersey	East	Harold Matthews	2016-06-02	Junk	3	12.42	37.26
12	Frances Warren	New York	East	Roy Young	2016-06-02	Widgets	8	53.35	426.8
13	Sara Snyder	New York	East	Debra Allen	2016-02-20	Things	1	17.83	17.83
14	Randy Watson	Connecticut	East	Alan Dean	2016-06-07	Junk	7	12.42	86.94
15	Randy Watson	Massachusetts	East	Robin Matthews	2016-10-31	Stuff	5	16.32	81.6
16	Randy Watson	New York	East	Randy Burton	2016-03-13	Stuff	4	16.32	65.28
17	Patrick Graham	Washington	West	Terry Nguyen	2016-02-10	Widgets	10	53.35	533.5

SQL

- **S**tructured **Q**uery **L**anguage
- Standard language for relational data models and relational databases
  - Relations are stored as tables
    - Columns are attributes, rows are tuples
- Data Definition Language (DDL)
  - Create/alter/delete tables and their attributes (columns)
- Data Manipulation Language (DML)
  - Query (search for data) in one or more tables
  - Insert/delete/update rows in tables

Data Types in SQL

- Atomic types
  - Characters: CHAR(20), VARCHAR(50)
  - Numbers: INT (INTEGER), BIGINT, SMALLINT, FLOAT (REAL), DOUBLE, DECIMAL (NUMERIC)
  - Others: MONEY, DATETIME, …
- Every column must have an atomic type
  - Tables are flat, cells are not merged or divided
  - Note: objects can be stored as JSON and XML (text-based)

Defining tables

CREATE TABLE *table_name* (
    column1 datatype [constraint],
    column2 datatype [constraint],
    column3 datatype [constraint],
    ....
);

```
CREATE TABLE Persons (
    ID int NOT NULL,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age int,
    PRIMARY KEY (ID)
);
```

Inserting data

INSERT INTO *table_name* (*column1*, *column2*, *column3*, ...)
VALUES (*value1*, *value2*, *value3*, ...);

```
INSERT INTO Persons (ID, LastName, FirstName)
VALUES (101, 'Smith', 'Adam'), (102, 'Smith','Eva');
```

Querying data

SELECT *column1*, *column2, ...* FROM *table_name* WHERE *conditions*;

```
SELECT * FROM Persons WHERE LastName = 'Smith'
```

Try SQL online: https://www.w3schools.com/sql/trysql.asp?filename=trysql_create_table

Updating data

UPDATE *table_name*
SET *column1 = value1, column2 = value2,* ...
WHERE *condition*;

```
UPDATE Persons SET Age = 20 WHERE ID = 101;
```

Deleting data

DELETE FROM *table_name* WHERE *condition*;

```
DELETE FROM Persons WHERE ID = 101;
```

Quiz: Analyze data in the sales management sample

	A	B	C	D	E	F	G	H	I
1	Sales Representative	Location	Region	Customer	Order Date	Item	Quantity	Price	Total Sale Amount
2	Sara Snyder	New York	East	Phyllis Johnston	2016-10-30	Things	1	17.83	17.83
3	Sara Snyder	New York	East	Kimberly Little	2016-05-23	Junk	3	12.42	37.26
4	Frances Warren	Massachusetts	East	Justin Dixon	2016-09-27	Widgets	4	53.35	213.4
5	Sara Snyder	Massachusetts	East	Shirley Rivera	2016-02-12	Junk	5	12.42	62.1
6	Diane Gonzalez	Oregon	West	Marilyn Franklin	2016-02-14	Things	8	17.83	142.64
7	Patrick Graham	Washington	West	Henry Sanders	2016-04-11	Widgets	4	53.35	213.4
8	Sara Snyder	Connecticut	East	Benjamin Phillips	2016-09-02	Junk	4	12.42	49.68
9	Frances Warren	New Jersey	East	Theresa Torres	2016-11-26	Junk	4	12.42	49.68
10	Patrick Graham	Oregon	West	Roger Bell	2016-07-13	Junk	10	12.42	124.2
11	Sara Snyder	New Jersey	East	Harold Matthews	2016-06-02	Junk	3	12.42	37.26
12	Frances Warren	New York	East	Roy Young	2016-06-02	Widgets	8	53.35	426.8
13	Sara Snyder	New York	East	Debra Allen	2016-02-20	Things	1	17.83	17.83
14	Randy Watson	Connecticut	East	Alan Dean	2016-06-07	Junk	7	12.42	86.94
15	Randy Watson	Massachusetts	East	Robin Matthews	2016-10-31	Stuff	5	16.32	81.6
16	Randy Watson	New York	East	Randy Burton	2016-03-13	Stuff	4	16.32	65.28
17	Patrick Graham	Washington	West	Terry Nguyen	2016-02-10	Widgets	10	53.35	533.5

https://www.w3schools.com/sql/trysql.asp?filename=trysql_create_table

Query components

SELECT: columns, expressions, functions

FROM: tables, views, sub-queries

WHERE: conditions on rows

ORDER BY: sorting results by columns' values

GROUP BY: grouping rows into groups for aggregation

HAVING: conditions on groups

DISTINCT: remove duplicates

LIMIT: select just a number of rows

Columns in SELECT

- To avoid confusion, columns can be referred via their tables
  - Example: SELECT Products.Name FROM Products, Customers...
- Columns, expressions, tables, queries... can be named (or renamed) with keyword AS
  - Example: SELECT sum(Total) AS AllTotal FROM Orders

Functions in SELECT

- Aggregation: Count, Sum, Avg, Min, Max
- Date functions: DATE(), YEAR(), MONTH(), DAY()
- String functions: UPPER(), LOWER(), CONCAT(), SUBSTRING(), LENGTH(), REPLACE(), TRIM()
- Math functions: ROUND(), ABS(), SQRT()
- More: https://www.w3schools.com/sql/sql_ref_mysql.asp

Conditions in SELECT

- Expressions can contains columns, functions, and sub-queries
- All typical math and logic operators:
  - +, -, *, /, <, >, <>, !=... BETWEEN x AND y
  - Logic: AND, OR, NOT
- Set operator
    - IN (belong to a set)
    - ALL (all elements), ANY (some elements),
    - EXISTS (set): check if set is empty
- String pattern matching: LIKE <pattern>.
    - % for a substring, _ for a character
    - LIKE '%Apple%': Containing the sub-string 'Apple'
    - LIKE 'iPhone_' : Contain a character after 'iPhone'

Example queries

1. Find all products with price less than 1$
2. Find all product names starting with 'Apple'
3. List top 10 most expensive products

4. Find all products that have not been sold any time

Grouping and calculating on groups

- Example: Calculating total sales of products

```
SELECT ProductID, sum(Quantity), sum(TotalCost)
FROM OrderLines
GROUP BY ProductID
```

Grouping and calculating on groups

- Example: Find product sold for at least 1000 units

```
SELECT ProductID, sum(Quantity) as TotalSoldUnit FROM OrderDetails
GROUP BY ProductID
HAVING TotalSoldUnit >= 1000
```

Grouping and calculating on groups

WHERE, GROUP BY, HAVING can be used as the same time

Example: Find product with price >= $20 and sold >= 10 times

Sub-query

- A query could be used in another query (sub-query)
  - Used in IN, ALL, ANY, or EXISTS operations
  - Used as a table in FROM
  - Used as an expression in SELECT

- Example: Find products' names bought at least 10 times

Get data from multiple tables - JOIN

SELECT FROM Table1, Table2, ... WHERE <joining condition>

Example: List names of products bought

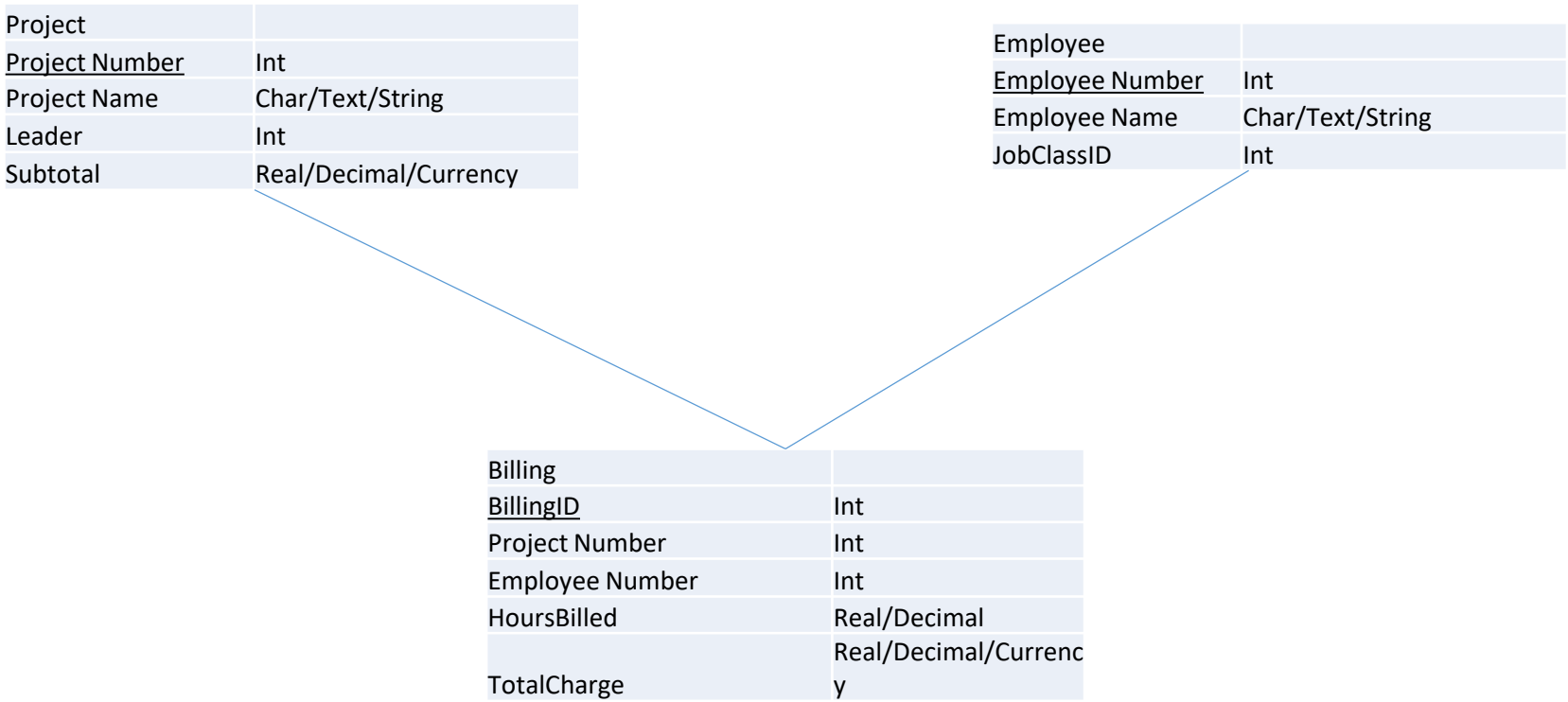
SELECT ProductID, Name FROM Products, Purchases

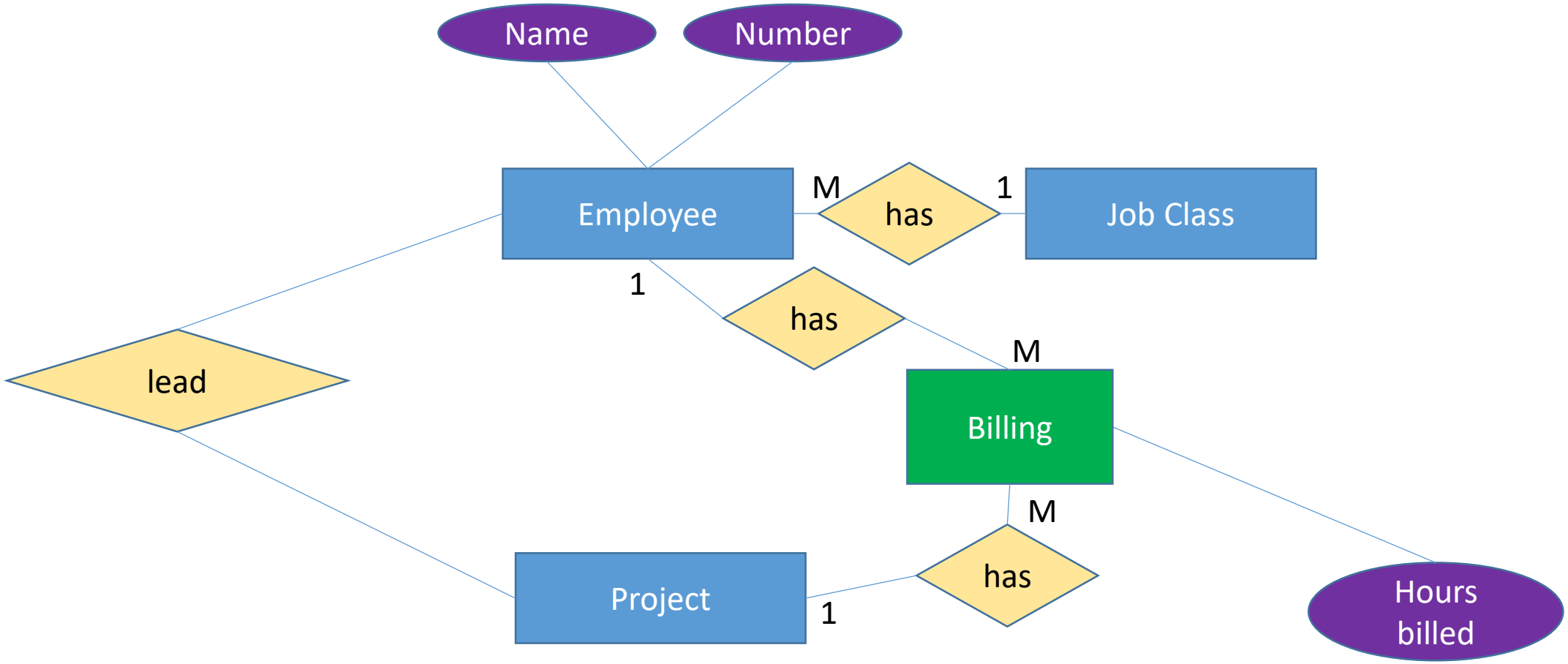
WHERE Products.ProductID = Purchases.ProductID

SELECT ProductID, Name FROM Products JOIN Purchases ON Products.ProductID = Purchases.ProductID

More example

- Find customers who did not buy any 'iPhone' products?
- Find products which are sold only once?
- Find products which are not sold yet?
- Find products which are sold in each purchase with sales more than total sales of any 'iPhone' products?


Project
Project Number
Int
Project Name
Char/Text/String
Leader
Int
Subtotal
Real/Decimal/Currency
Employee
Employee Number
Int
Employee Name
Char/Text/String
JobClassID
Int
Billing
BillingID
Int
Project Number
Int
Employee Number
Int
HoursBilled
Real/Decimal
TotalCharge
Real/Decimal/Currency


Name
Number
Employee
M
has
1
Job Class
1
has
M
lead
Billing
M
has
Project
1
Hours billed