2020-2021

5COSC002W DATABASE SYSTEMS
Lecture 05

DATABASE IMPLEMENTATION
SQL: Creating database tables

UNIVERSITY OF WESTMINSTER

UNIVERSITY OF TECHNOLOGICAL INNOVATION WESTMINSTER

Lecture 05 – Outline

– *Relational Model*
  - Tables and properties
  - SKs, CKs, FKs, PKs
  - Entity integrity & referential integrity

– *RDBMSs & SQL*
  - Oracle vs. MySQL
  - SQL = DDL + DML + DCL + Transaction control

– *DDL: Creating tables in SQL (in MySQL DBMS)*
  - Data types
  - Constraints

– *DML: Manipulating records in SQL (in MySQL DBMS)*
  - Inserting records
  - Updating & deleting records

Database Design Methodology – Step 3

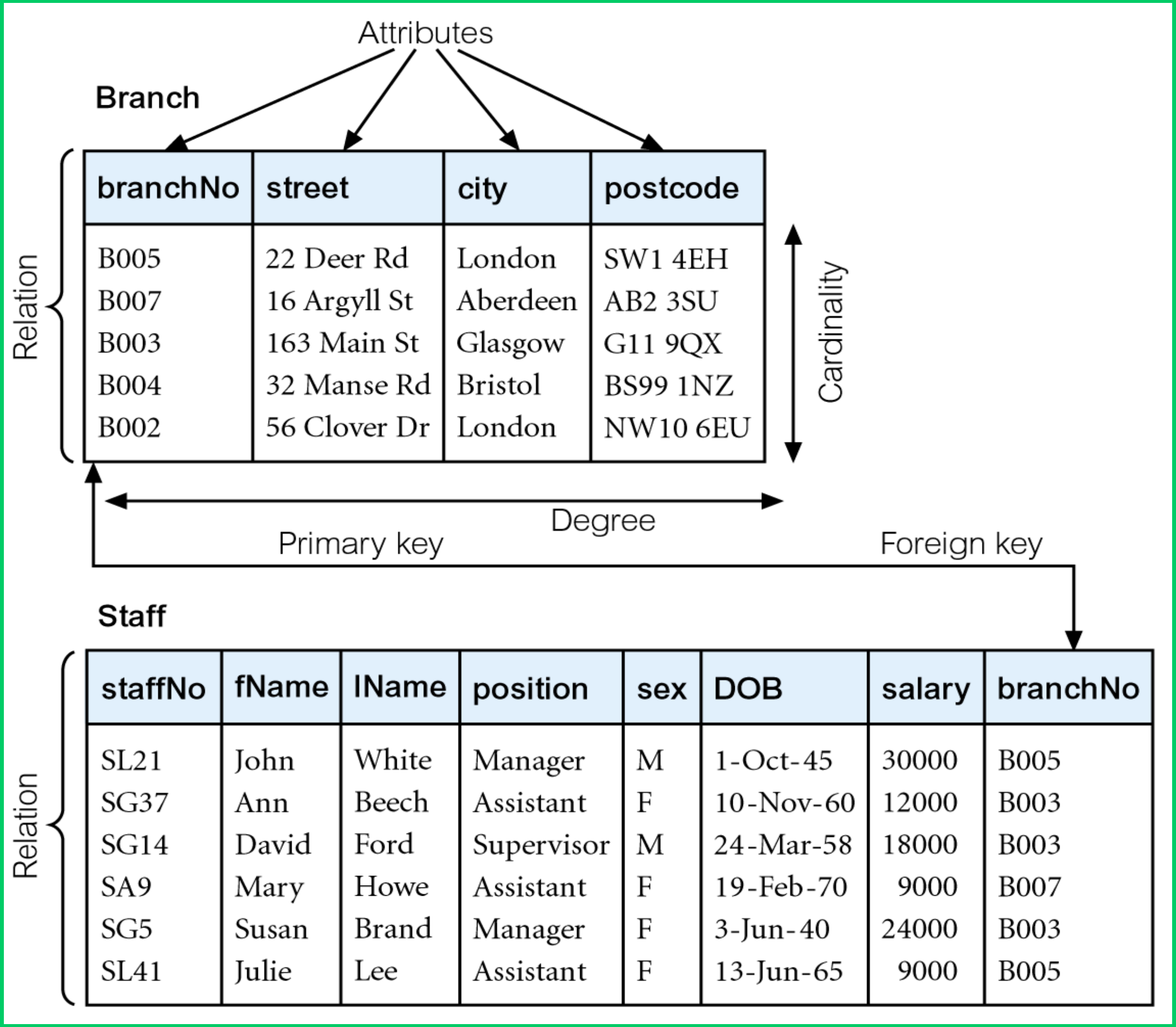
PHYSICAL DESIGN & IMPLEMENTATION

- **Translate logical data model for target DBMS**
  - Step 3.1 Create base relations in SQL & document
  - Step 3.2 Create representation of derived data in SQL
  - Step 3.3 Create constraints in SQL
- **Design file organizations and indexes**
  - Step 3.4 Analyze transactions
  - Step 3.5 Choose file organizations
  - Step 3.6 Choose indexes
  - Step 3.7 Estimate disk space requirements
- **Design user views & security mechanisms**
  - Step 3.8 Design and create user views
  - Step 3.9 Design and implement security mechanisms

UNIVERSITY OF WESTMINSTER

Relational Model

Interconnected relations



Branch

branchNo	street	city	postcode
B005	22 Deer Rd	London	SW1 4EH
B007	16 Argyll St	Aberdeen	AB2 3SU
B003	163 Main St	Glasgow	G11 9QX
B004	32 Manse Rd	Bristol	BS99 1NZ
B002	56 Clover Dr	London	NW10 6EU

Staff

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000	B005

Relational Model Terminology

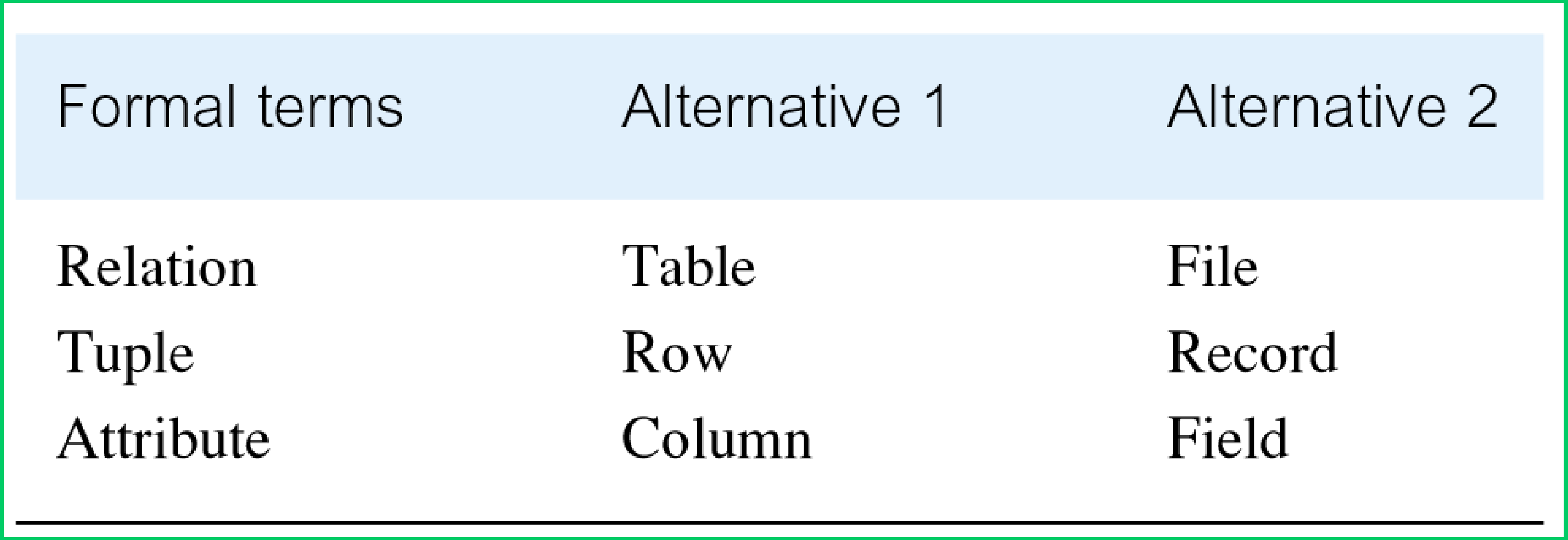
- A **RELATION** is a table with columns and rows.
  - Only applies to logical structure of the database, not the physical structure.

- **ATTRIBUTE** is a named column of a relation.
- **DOMAIN** is the set of allowable values for one or more attributes.
- **DEGREE** is the number of attributes in a relation.

- **TUPLE** is a row of a relation.
- **CARDINALITY** is the number of tuples in a relation.

UNIVERSITY OF TECHNOLOGICAL INNOVATION WESTMINSTER

Relational Model Alternative Terminology



Formal terms	Alternative 1	Alternative 2
Relation	Table	File
Tuple	Row	Record
Attribute	Column	Field

UNIVERSITY OF TECHNOLOGICAL INNOVATION WESTMINSTER

Relational Database & Relational Schema

- **RELATIONAL DATABASE**
  - Collection of normalized relations with distinct relation names.
- **RELATION SCHEMA**
  - Named relation defined by a set of attribute and domain name pairs.
- **RELATIONAL DATABASE SCHEMA**
  - Set of relation schemas, each with a distinct name.

UNIVERSITY OF
TECHNOLOGICAL
INNOVATION
WESTMINSTER

Properties of Relations

- Relation name is distinct from all other relation names in relational schema.
- Each cell of relation contains exactly one atomic (single) value.

- Each attribute has a distinct name.
- Values of an attribute are from the same domain.
- Order of attributes has no significance.

- Each tuple is distinct; no duplicate tuples.
- Order of tuples has no significance, theoretically.

UNIVERSITY OF TECHNOLOGICAL INNOVATION WESTMINSTER

Relational Keys

- **Superkey**
  - An attribute, or set of attributes, that uniquely identifies a tuple within a relation.

- **Candidate Key**
  - Superkey (K) such that no proper subset is a superkey within the relation.
  - In each tuple of R, values of K uniquely identify that tuple (uniqueness).
  - No proper subset of K has the uniqueness property (irreducibility).

UNIVERSITY OF TECHNOLOGICAL INNOVATION WESTMINSTER

Relational Keys

- **Primary Key**
  - Candidate key selected to identify tuples uniquely within relation.
- **Alternate Keys**
  - Candidate keys that are not selected to be primary key.
- **Foreign Key**
  - Attribute, or set of attributes, within one relation that matches candidate key of some (possibly same) relation.

UNIVERSITY OF TECHNOLOGICAL INNOVATION WESTMINSTER

Null Values

- **Null**
  - Represents value for an attribute that is currently unknown or not applicable for tuple.
  - Deals with incomplete or exceptional data.
  - Represents the absence of a value and is not the same as zero or spaces, which are values.

UNIVERSITY OF TECHNOLOGICAL INNOVATION WESTMINSTER

Integrity Constraints & General Constraints

– **Entity Integrity**
  - In a base relation, no attribute of a primary key can be null.

– **Referential Integrity**
  - If foreign key exists in a relation, either foreign key value must match a candidate key value of some tuple in its home relation or foreign key value must be wholly null.

– **General Constraints**
  - Additional rules specified by users or database administrators that define or constrain some aspect of the enterprise.

RDBMSs: Oracle vs. MySQL

– **Oracle**
  - Historically first commercial RDBMS based on SQL.
  - ORDBMS: O.O. extensions
  - DDBMS: distributed functionality
  - Oracle Internet Platform based on standards such as:
    - HTTP and HTML/XML
    - Java, J2EE, SQLJ, JSP
    - Web services: SOAP, WDSL, UDDI, etc.

– **MySQL** **(not to be confused with SQL: Structured Query Language)**
  - Open-source RDBMS now owned by Oracle Corporation
  - Central component of the LAMP software stack: Linux, Apache, MySQL, Perl/PHP/Python
  - Used in applications such as Joomla, WordPress, Drupal.
  - Used in sites such as Facebook, Twitter, Flicker, YouTube.

UNIVERSITY OF TECHNOLOGICAL INNOVATION WESTMINSTER

SQL Statements

SELECT
INSERT
UPDATE
DELETE
MERGE
	Data Manipulation Language (DML)

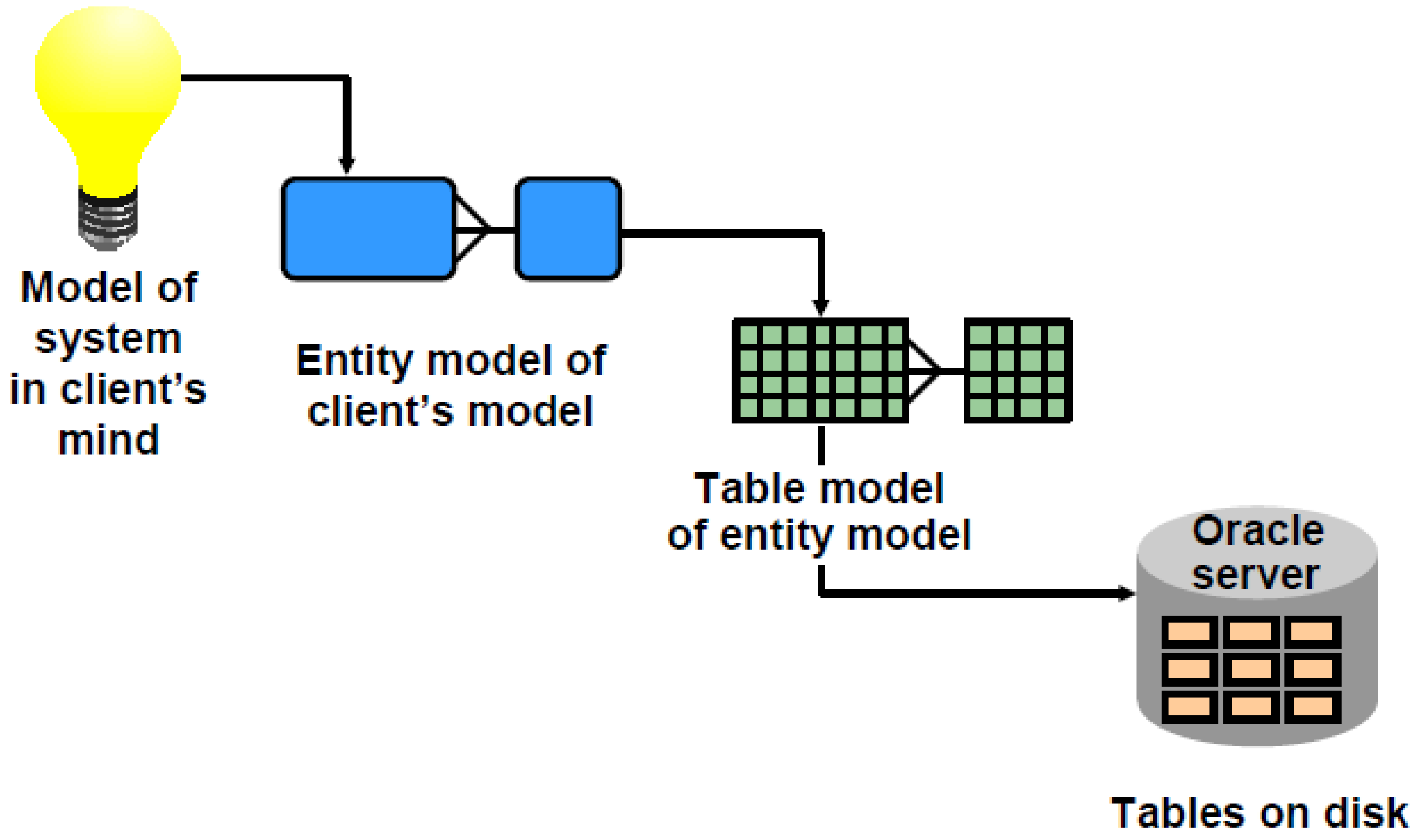
CREATE ALTER DROP RENAME TRUNCATE COMMENT	Data Definition Language (DDL)
GRANT REVOKE	Data control Language (DCL)
COMMIT ROLLBACK SAVEPOINT	Transaction Control

UNIVERSITY OF TECHNOLOGICAL INNOVATION WESTMINSTER

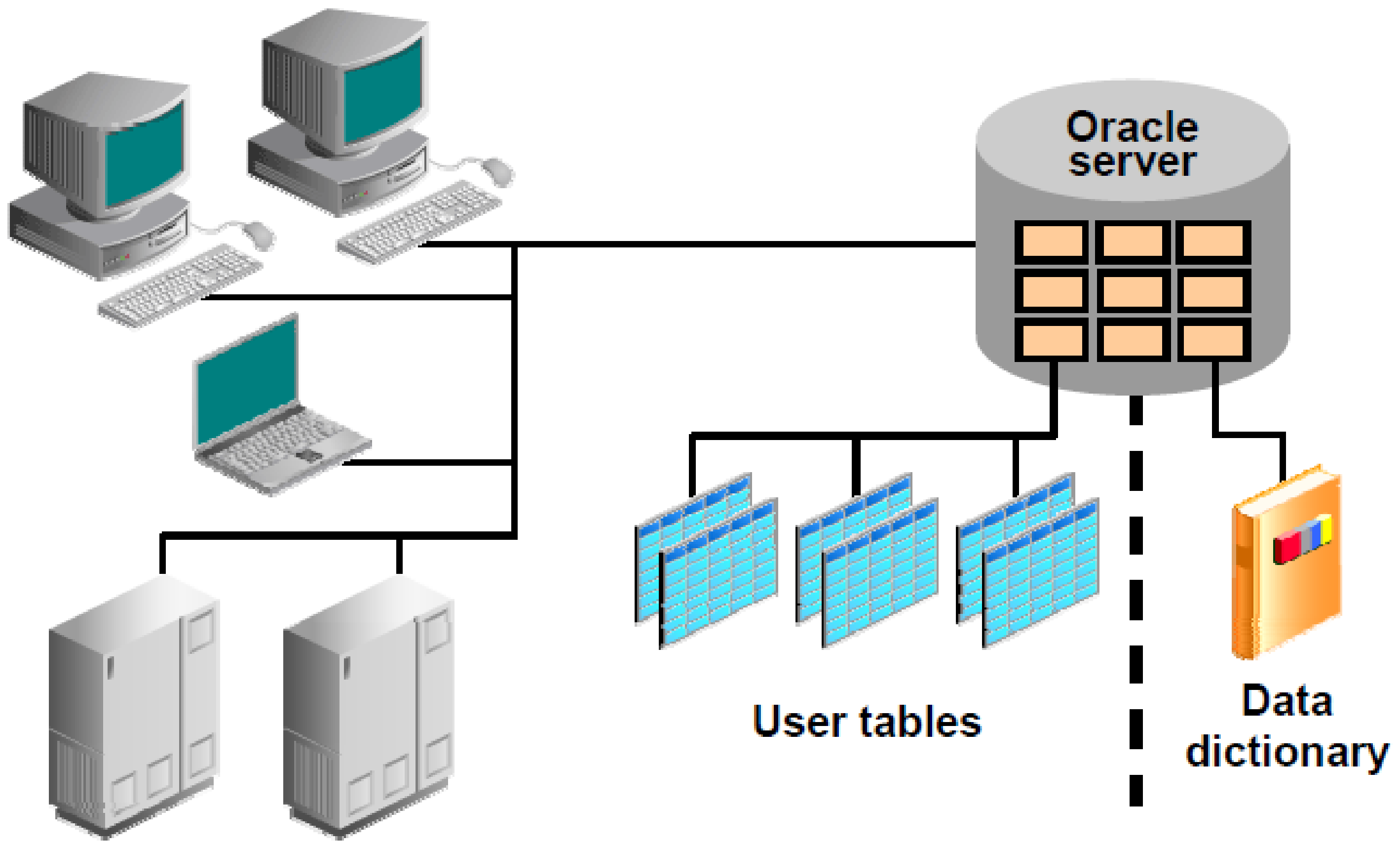
Database Objects

	Oracle	MySQL
Table / Relation	Basic unit of storage; composed of rows	Basic unit of storage; composed of rows
View	Subsets of data from one or more tables	Subsets of data from one or more tables
Index	Improves performance of some queries	Improves performance of some queries
Sequence	Generates numeric values	Not a separate object, use AUTO_INCREMENT
Synonym	Gives alternative names to objects	Not a separate object

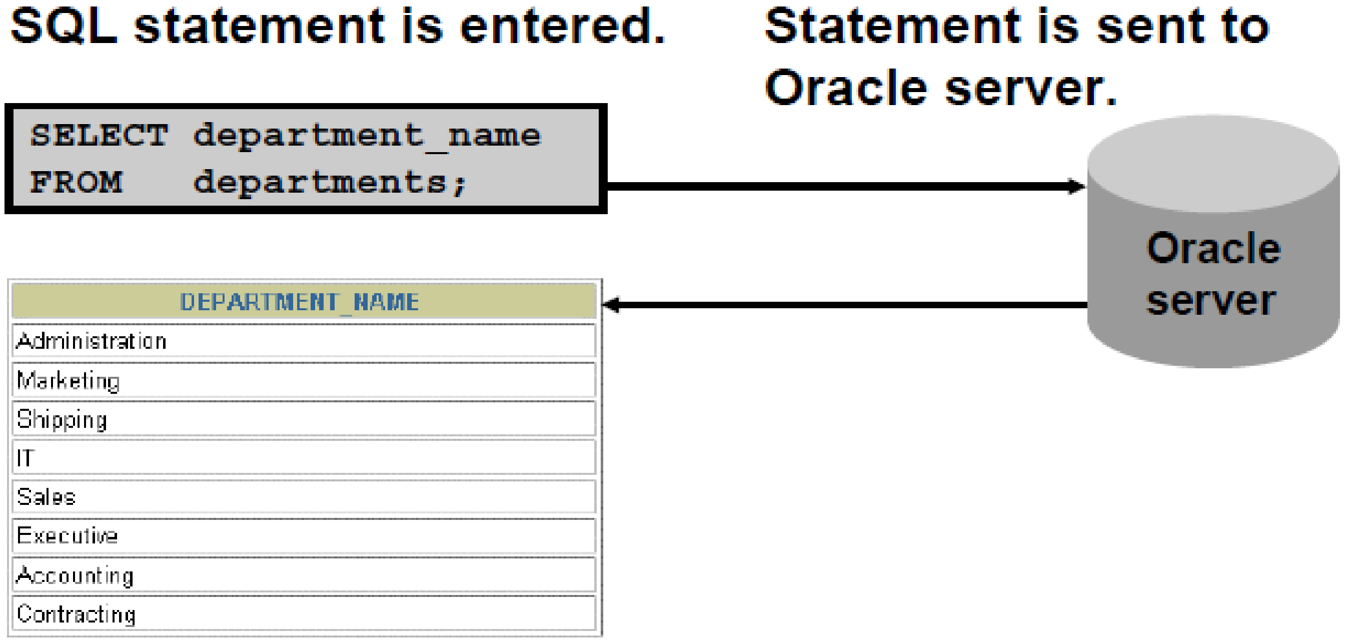
Oracle: Data Models


Model of system in client's mind
Entity model of client's model
Table model of entity model
Oracle server
Tables on disk

Oracle's Relational Database Management System



Communicating with an RDBMS Using SQL



SQL statement is entered.

```
SELECT department_name
FROM   departments;
```

Statement is sent to Oracle server.

DEPARTMENT_NAME
Administration
Marketing
Shipping
IT
Sales
Executive
Accounting
Contracting

SQL Statements

`SELECT`
`INSERT`
`UPDATE`
`DELETE`
`MERGE`
	Data manipulation language (DML)

`CREATE`
`ALTER`
`DROP`
`RENAME`
`TRUNCATE`
`COMMENT`
	Data definition language (DDL)

`GRANT` `REVOKE`	Data control language (DCL)

`COMMIT`
`ROLLBACK`
`SAVEPOINT`
	Transaction control

Oracle: Data Types

Data Type	Description
`VARCHAR2(`*`size`*`)`	Variable-length character data
`CHAR(`*`size`*`)`	Fixed-length character data
`NUMBER(`*`p,s`*`)`	Variable-length numeric data
`DATE`	Date and time values
`LONG`	Variable-length character data (up to 2 GB)
`CLOB`	Character data (up to 4 GB)
`RAW and LONG RAW`	Raw binary data
`BLOB`	Binary data (up to 4 GB)
`BFILE`	Binary data stored in an external file (up to 4 GB)
`ROWID`	A base-64 number system representing the unique address of a row in its table

Oracle: Including Constraints

- Constraints enforce rules at the table level.
- Constraints prevent the deletion of a table if there are dependencies.
- The following constraint types are valid:
  - `NOT NULL`
  - `UNIQUE`
  - `PRIMARY KEY`
  - `FOREIGN KEY`
  - `CHECK`

UNIVERSITY OF TECHNOLOGICAL INNOVATION WESTMINSTER

MySQL: Data Types (Text)

Data type	Description
CHAR(size)	Holds a fixed length string (can contain letters, numbers, and special characters). The fixed size is specified in parenthesis. Can store up to 255 characters
VARCHAR(size)	Holds a variable length string (can contain letters, numbers, and special characters). The maximum size is specified in parenthesis. Can store up to 255 characters. **Note:** If you put a greater value than 255 it will be converted to a TEXT type
TINYTEXT	Holds a string with a maximum length of 255 characters
TEXT	Holds a string with a maximum length of 65,535 characters
BLOB	For BLOBs (Binary Large OBjects). Holds up to 65,535 bytes of data
MEDIUMTEXT	Holds a string with a maximum length of 16,777,215 characters
MEDIUMBLOB	For BLOBs (Binary Large OBjects). Holds up to 16,777,215 bytes of data
LONGTEXT	Holds a string with a maximum length of 4,294,967,295 characters
LONGBLOB	For BLOBs (Binary Large OBjects). Holds up to 4,294,967,295 bytes of data
ENUM(x,y,z,etc.)	Let you enter a list of possible values. You can list up to 65535 values in an ENUM list. If a value is inserted that is not in the list, a blank value will be inserted. **Note:** The values are sorted in the order you enter them. You enter the possible values in this format: ENUM('X','Y','Z')
SET	Similar to ENUM except that SET may contain up to 64 list items and can store more than one choice

UNIVERSITY OF TECHNOLOGICAL INNOVATION WESTMINSTER

MySQL: Data Types (Number)

Data type	Description
TINYINT(size)	-128 to 127 normal. 0 to 255 UNSIGNED*. The maximum number of digits may be specified in parenthesis
SMALLINT(size)	-32768 to 32767 normal. 0 to 65535 UNSIGNED*. The maximum number of digits may be specified in parenthesis
MEDIUMINT (size)	-8388608 to 8388607 normal. 0 to 16777215 UNSIGNED*. The maximum number of digits may be specified in parenthesis
INT(size)	-2147483648 to 2147483647 normal. 0 to 4294967295 UNSIGNED*. The maximum number of digits may be specified in parenthesis
BIGINT(size)	-9223372036854775808 to 9223372036854775807 normal. 0 to 18446744073709551615 UNSIGNED*. The maximum number of digits may be specified in parenthesis
FLOAT(size,d)	A small number with a floating decimal point. The maximum number of digits may be specified in the size parameter. The maximum number of digits to the right of the decimal point is specified in the d parameter
DOUBLE(size,d)	A large number with a floating decimal point. The maximum number of digits may be specified in the size parameter. The maximum number of digits to the right of the decimal point is specified in the d parameter
DECIMAL(size,d)	A DOUBLE stored as a string , allowing for a fixed decimal point. The maximum number of digits may be specified in the size parameter. The maximum number of digits to the right of the decimal point is specified in the d parameter

UNIVERSITY OF TECHNOLOGICAL INNOVATION WESTMINSTER

MySQL: Data Types (Date)

Data type	Description
DATE()	A date. Format: YYYY-MM-DD ***Note:*** *The supported range is from '1000-01-01' to '9999-12-31'*
DATETIME()	*A date and time combination. Format: YYYY-MM-DD HH:MI:SS ***Note:*** *The supported range is from '1000-01-01 00:00:00' to '9999-12-31 23:59:59'*
TIMESTAMP()	*A timestamp. TIMESTAMP values are stored as the number of seconds since the Unix epoch ('1970-01-01 00:00:00' UTC). Format: YYYY-MM-DD HH:MI:SS ***Note:*** *The supported range is from '1970-01-01 00:00:01' UTC to '2038-01-09 03:14:07' UTC*
TIME()	A time. Format: HH:MI:SS ***Note:*** *The supported range is from '-838:59:59' to '838:59:59'*
YEAR()	A year in two-digit or four-digit format. ***Note:*** *Values allowed in four-digit format: 1901 to 2155. Values allowed in two-digit format: 70 to 69, representing years from 1970 to 2069*

**Even if DATETIME and TIMESTAMP return the same format, they work very differently.*

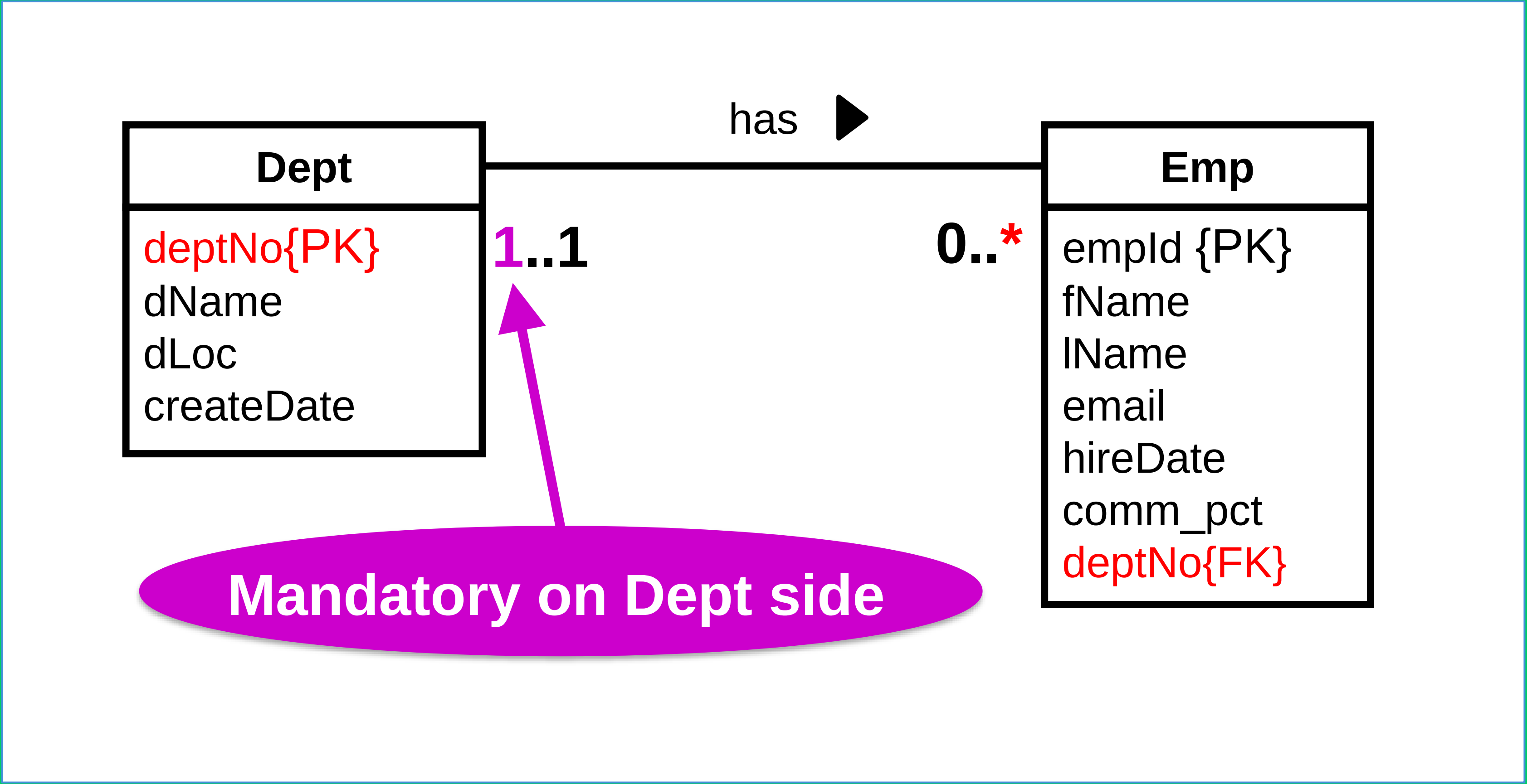
In an INSERT or UPDATE query, the TIMESTAMP automatically set itself to the current date and time.

TIMESTAMP also accepts various formats, like YYYYMMDDHHMISS, YYMMDDHHMISS, YYYYMMDD, or YYMMDD.

UNIVERSITY OF TECHNOLOGICAL INNOVATION WESTMINSTER

DDL: Creating Tables in MySQL (1)

– Logical ERD



UNIVERSITY OF TECHNOLOGICAL INNOVATION WESTMINSTER

DDL: Creating Dept Table in MySQL(1)

```
CREATE TABLE Dept
(
	deptNo		INT(4),
	dName		VARCHAR(20) UNIQUE NOT NULL,
	dLoc		VARCHAR(30) NOT NULL,
	createDate TIMESTAMP,
	constraint	d_dno_pk PRIMARY KEY (deptNo)
);
```

UNIVERSITY OF TECHNOLOGICAL INNOVATION

DDL: Creating Emp Table in MySQL(1)

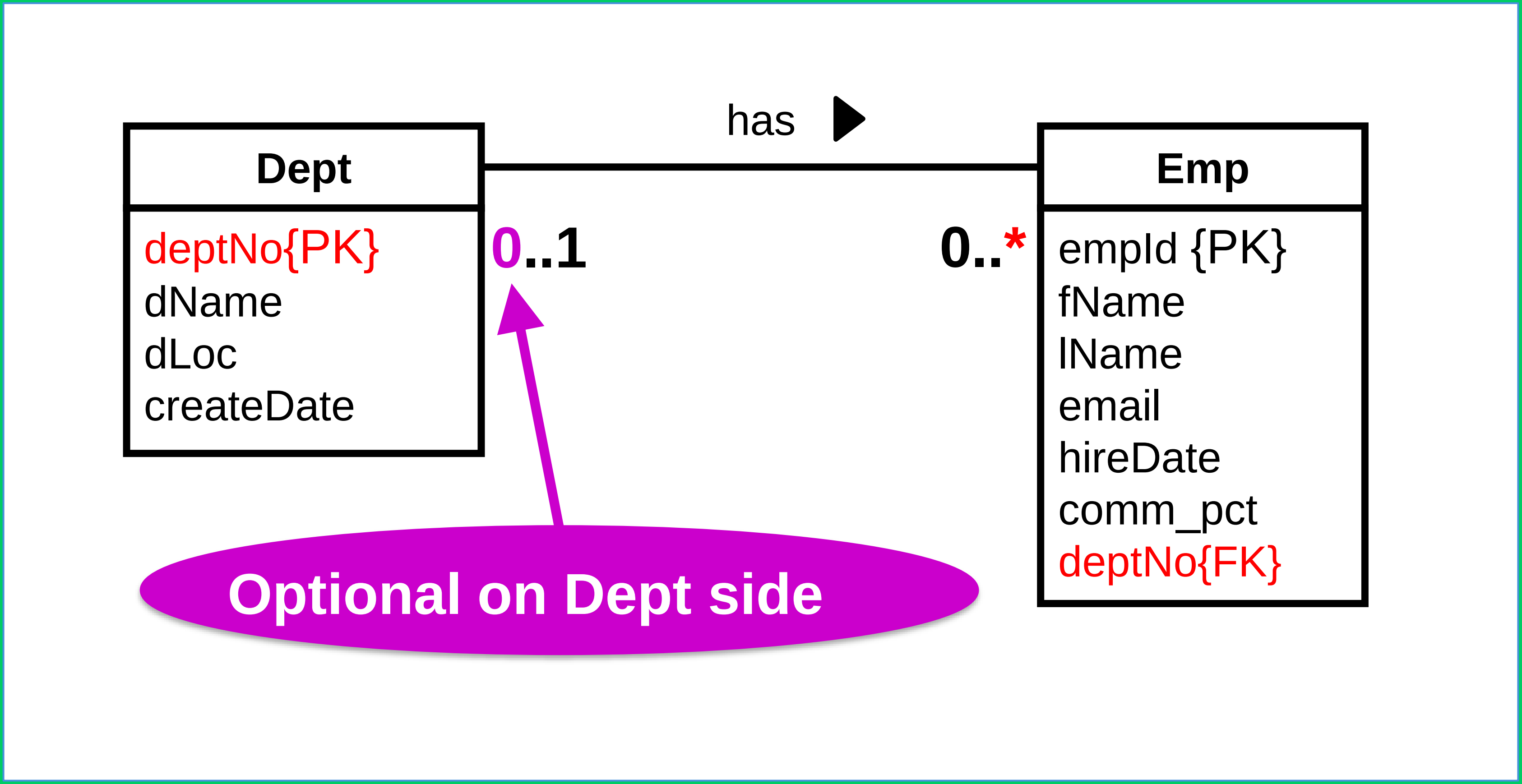
```
CREATE TABLE Emp
(
        empId       INT(6),
        fName       VARCHAR(50) NOT NULL,
        lName       VARCHAR(50) NOT NULL,
        email       VARCHAR(100) UNIQUE NOT NULL,
        hireDate    DATE,
        comm_pct    DECIMAL(2,2),
        deptNo      INT(4) NOT NULL,
        constraint  e_eid_pk PRIMARY KEY (empId),
        constraint  e_dno_fk FOREIGN KEY (deptNo)
        references  Dept(deptNo)
);
```

***Not Null Constraint* needed**

UNIVERSITY OF WESTMINSTER

DDL: Creating Tables in MySQL (2)

– Logical ERD



UNIVERSITY OF WESTMINSTER

DDL: Creating Dept Table in MySQL(2)

```
CREATE TABLE Dept
(
        deptNo      INT(4),
        dName       VARCHAR(20) UNIQUE NOT NULL,
        dLoc        VARCHAR(30) NOT NULL,
        createDate  TIMESTAMP,
        constraint  d_dno_pk PRIMARY KEY (deptNo)
);
```

UNIVERSITY OF TECHNOLOGICAL INNOVATION

DDL: Creating Emp Table in MySQL(2)

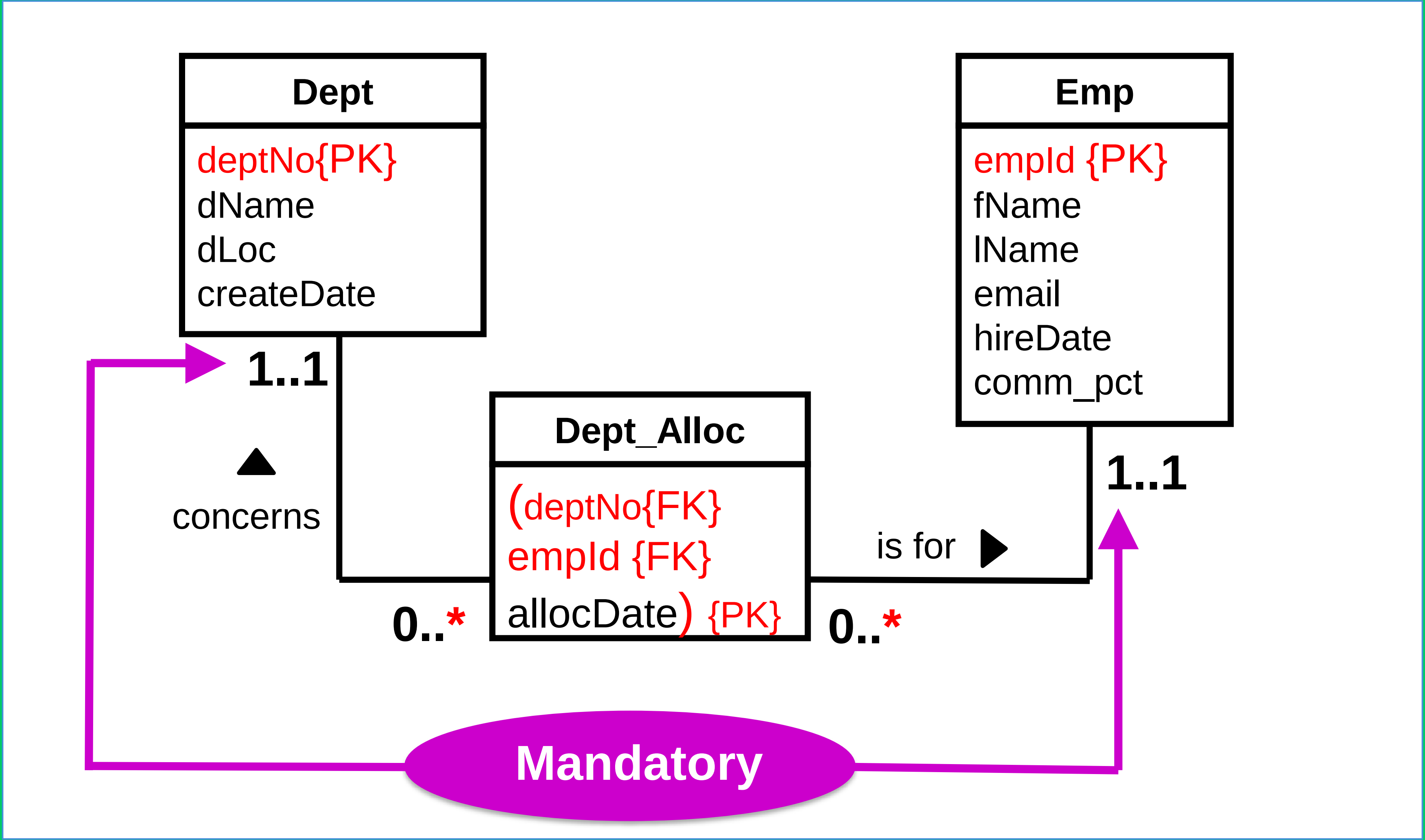
```
CREATE TABLE Emp
(
        empId         INT(6),
        fName         VARCHAR(50) NOT NULL,
        lName         VARCHAR(50) NOT NULL,
        email         VARCHAR(100) UNIQUE NOT NULL,
        hireDate      DATE,
        comm_pct      DECIMAL(2,2),
        deptNo        INT(4),
        constraint    e_eid_pk PRIMARY KEY (empId),
        constraint    e_dno_fk FOREIGN KEY (deptNo)
        references    Dept(deptNo)
);
```

***Not Null Constraint* NOT needed**

UNIVERSITY OF TECHNOLOGICAL INNOVATION WESTMINSTER

DDL: Creating Tables in MySQL (3)

– Logical ERD



UNIVERSITY OF WESTMINSTER

DDL: Creating Dept Table in MySQL(3)

```
CREATE TABLE Dept
(
        deptNo      INT(4),
        dName       VARCHAR(20) UNIQUE NOT NULL,
        dLoc        VARCHAR(30) NOT NULL,
        createDate TIMESTAMP,
        constraint  d_dno_pk PRIMARY KEY (deptNo)
);
```

UNIVERSITY OF TECHNOLOGICAL INNOVATION WESTMINSTER

DDL: Creating Emp Table in MySQL(3)

```
CREATE TABLE Emp
(
        empId       INT(6),
        fName       VARCHAR(50) NOT NULL,
        lName       VARCHAR(50) NOT NULL,
        email       VARCHAR(100) UNIQUE NOT NULL,
        hireDate    DATE,
        comm_pct    DECIMAL(2,2),
        constraint  e_eid_pk PRIMARY KEY (empId)
);
```

DDL: Creating Dept_Alloc Table (3)

```
CREATE TABLE Dept_Alloc
(
        deptNo      INT(4) NOT NULL,
        empId       INT(6) NOT NULL,
        allocDate   TIMESTAMP,
        constraint  d_composite_pk PRIMARY KEY
        (deptNo, empId, allocDate),
        constraint  da_dno_fk FOREIGN KEY (deptNo)
        references  Dept(deptNo),
        constraint  da_eid_fk FOREIGN KEY (empId)
        references  Emp(empId)
);
```

***Not Null Constraints* needed**

UNIVERSITY OF TECHNOLOGICAL INNOVATION WESTMINSTER

DDL: Dropping Tables in MySQL

```
DROP TABLE Emp;
DROP TABLE Dept;
```

- **All data and structure in the table are deleted**
- **Any pending transactions are committed**
- **All indexes are dropped**
- **All constraints are dropped**
- **You cannot roll back**
- **Drop Child table before Parent table otherwise referential integrity is breached**

DDL: Truncate Tables in MySQL

```
TRUNCATE TABLE Emp;
TRUNCATE TABLE Dept;
```

- It removes all rows from a table
- It leaves the table empty
- It leaves the table structure intact
- It cannot be easily undone
- Truncate Child table before Parent table otherwise referential integrity is breached

UNIVERSITY OF TECHNOLOGICAL INNOVATION WESTMINSTER

DDL: Rename Tables in MySQL

```
RENAME TABLE Emp to Employee;
RENAME TABLE Dept to Department;
```

- **It changes the name of the table**

- **It is equivalent to:**

```
ALTER TABLE Emp RENAME Employee;
ALTER TABLE Dept RENAME Department;
```

DDL: Altering Columns in MySQL

```
ALTER TABLE Emp ADD dateOfBirth date;

ALTER TABLE Emp
CHANGE COLUMN dateOfBirth DOB date;

ALTER TABLE Emp
CHANGE COLUMN DOB DOB year;

ALTER TABLE Emp
CHANGE COLUMN DOB DOB year not null;

ALTER TABLE Emp DROP COLUMN DOB;
```

UNIVERSITY OF TECHNOLOGICAL INNOVATION WESTMINSTER

DML: Insert records in tables in MySQL

```
INSERT INTO
Dept (deptNo, dName, dLoc)
VALUES (10, 'IT', 'London');

INSERT INTO
Dept (deptNo, dName, dLoc)
VALUES (20, 'Marketing', 'Brighton');
```

- **Insert new row containing values for each column**
- **List values in default order of columns in table**
- **Single quotes ' ' for text and dates, not numbers**

DML: Insert records in tables in MySQL

– **Date has to be in the correct MySQL format**

```
INSERT INTO
Emp (empId, fName, lName, email, hireDate,
commission_pct, deptNo)
VALUES (1, 'Joe', 'Bloggs', 'jb@mail.com',
'2016-10-15', 0.15, 20);
```

– **Listing column names after table is optional (but a very good idea!)**

```
INSERT INTO Emp
VALUES (2, 'Jen', 'Smith', 'js@mail.com',
'2016-10-02', 0.20, 10);
```

UNIVERSITY OF TECHNOLOGICAL INNOVATION WESTMINSTER

DML: Insert rows with NULLs in MySQL

– **Implicitly: omit the column from the column list**

```
INSERT INTO
Emp (empId, fName, lName, email)
VALUES (3, 'Jim', 'Green', 'jg@mail.com');
```

– **Explicitly: specify NULL in the VALUES clause**

```
INSERT INTO Emp
VALUES (4, 'Kim', 'Brown', 'kb@mail.com',
NULL, NULL, NULL);
```

DML: Update records in MySQL

– **Modify all rows in table**

```
UPDATE Emp
SET deptNo = 20;
```

– **Modify specific row(s) in table: add WHERE clause**

```
UPDATE Emp
SET deptNo = 10
WHERE empId = 1 or empId = 2;
```

UNIVERSITY OF TECHNOLOGICAL INNOVATION WESTMINSTER

DML: Delete records in MySQL

- **Delete all rows in table (like TRUNCATE)**

```
DELETE FROM Emp;
```

- **Delete specific row(s) in table: add WHERE clause**

```
DELETE FROM Emp
WHERE fname = 'Joe' or deptNo = 10;
```