**Introduction Lecture 1/21/20**

Attached Files:

* [[File](https://learn.bu.edu/bbcswebdav/pid-7673241-dt-content-rid-36483470_1/xid-36483470_1) Lecture 1.ipynb](https://learn.bu.edu/bbcswebdav/pid-7673241-dt-content-rid-36483470_1/xid-36483470_1) (8.469 KB)

Introduction to databases and database management systems.   Examples in class.

Attached file is a Jupyter notebook file.  Run Jupyter notebook to view it's contents.

Attached Jupiter Notebook

**Professors DB ER Diagram Problem 1/23/20**

Attached Files:

* [[File](https://learn.bu.edu/bbcswebdav/pid-7616914-dt-content-rid-36248815_1/xid-36248815_1) Professors DB design problem lecture 2.pdf](https://learn.bu.edu/bbcswebdav/pid-7616914-dt-content-rid-36248815_1/xid-36248815_1) (55.655 KB)
* [[File](https://learn.bu.edu/bbcswebdav/pid-7616914-dt-content-rid-36340976_1/xid-36340976_1) ER diagram problems for homework.pdf](https://learn.bu.edu/bbcswebdav/pid-7616914-dt-content-rid-36340976_1/xid-36340976_1) (50.228 KB)

Interpreting written descriptions of data and making ER diagrams.  
  
• Entity  
    • something described by attributes  
• Relationship  
    • something that links entities  
• Keys  
    • always one or more attributes  
    • for an entity  
    • for a relationship  
• Relationship classes  
    • one to many  
    • many to many  
    • one to one  
• Participation classes  
    • partial participation  
    • total participation  
• Relationship class and participation class notation  
    • triangle used in one to many relationship on the  
      side where only one line can be drawn  
    • circles used everywhere else  
    • total participation filled in circles and triangles  
• Ternary relationships  
• Aggregate (relationship treated like an entity)  
• Relationship attributes

**Create Table Professor SQL 1/28/20**

Attached Files:

* [[File](https://learn.bu.edu/bbcswebdav/pid-7616915-dt-content-rid-36248811_1/xid-36248811_1) Professor table schema and SQL and ER diagram.pdf](https://learn.bu.edu/bbcswebdav/pid-7616915-dt-content-rid-36248811_1/xid-36248811_1) (5.65 MB)
* [[File](https://learn.bu.edu/bbcswebdav/pid-7616915-dt-content-rid-36598808_1/xid-36598808_1) practice ER diagram problems.pdf](https://learn.bu.edu/bbcswebdav/pid-7616915-dt-content-rid-36598808_1/xid-36598808_1) (127.054 KB)

Creating tables from an ER diagram  
  
• Example DB table  
    • attribute, field, column (all refer to the same thing)  
    • record, tuple, row (all refer to the same thing)  
    • the KEY for every row in a table must be UNIQUE  
• Create Table statements  
    • list attributes  
    • give each field a data type  
    • NOT NULL -- for fields that MUST contain a value when a record is created  
        • all KEY fields must be NOT NULL  
    • AUTO\_INCREMENT -- typically for an integer key field in an ENTITY table  
    • ENUM -- when the allowed values are enumerated  
    • PRIMARY KEY() -- tells the system the KEY for the table (causes an INDEX to be built)  
    • ENGINE=INNODB -- tells system to use tables that support FOREIGN KEYS  
• Entity Table  
    • list attributes as defined in the ER diagram  
• Relationship Table  
    • many-to-many relationship  
        • create a new table  
        • list only the KEYS from the participating entities  
        • every KEY must be NOT NULL  
        • list any relationship attribute  
        • PRIMARY KEY is the combination of the KEYS from the participating entities  
        • FOREIGN KEY tells the system which table each part of the primary key comes from  
            • list for each attribute in the primary key  
            • key name in the relationship table doesn't have to match linked name in the entity table  
            • causes the system to check for matching values in the linked entity tables when entering data in the relationship table

### Creating Tables, Using Foreign Keys 1/30/20

• Relationship Table  
     • one-to-many relationship  
         • DO NOT CREATE A NEW TABLE  
         • modify the entity table on the triangle side of the ER diagram  
             • this entity table ALREADY CONTAINS THE RELATIONSHIP KEY  
         • add, AS AN ATTRIBUTE, the primary key from the other table participating in the relationship  
             • add NOT NULL to the new attribute IF THE MODIFIED ENTITY TABLE HAS FULL PARTICIPATION IN THE RELATIONSHIP  
         • add a FOREIGN KEY for the new attribute  
         • add any relationship attributes  
    • one-to-one relationship  
        • same as a one-to-many relationship  
        • either existing entity table can be modified  
            • choose the table which keeps relevant information together and wastes less memory space

Foreign Key statement functions  
• check that the VALUE entered in the foreign key field in the CHILD table matches a value in the PARENT table key field  
• apply the RULE STATED IN THE CHILD TABLE when deleting or updating in the PARENT table  
    • ON DELETE  
        • NO ACTION -- block the delete in the parent table if there is a matching value in the child table  
        • CASCADE -- allow the delete in the parent table and also delete rows with a matching value in the child table  
        • SET NULL -- allow the delete in the parent table and set matching values to NULL in the child table  
    • ON UPDATE  
        • NO ACTION -- block the update in the parent table if there is a matching value in the child table  
        • CASCADE -- allow the update in the parent table and also update to the new value in rows with a matching value in the child table  
        • SET NULL -- allow the update in the parent table and set matching values to NULL in the child table

### Introduction to Table Queries and Joins 2/4/20

Attached Files:

* + [[File](https://learn.bu.edu/bbcswebdav/pid-7712129-dt-content-rid-36921441_1/xid-36921441_1) Inserting and retrieving data, notes on joins.2.4.20.pdf](https://learn.bu.edu/bbcswebdav/pid-7712129-dt-content-rid-36921441_1/xid-36921441_1) (107.87 KB)
  + [[File](https://learn.bu.edu/bbcswebdav/pid-7712129-dt-content-rid-36998442_1/xid-36998442_1) Answer Notes on joins.2.4.20.pdf](https://learn.bu.edu/bbcswebdav/pid-7712129-dt-content-rid-36998442_1/xid-36998442_1) (91.818 KB)

Topics:  
  
    • Review of the protein function ER diagram  
    • SQL Select statement for single tables  
     
            SELECT...  
            FROM...  
            WHERE...  
            ORDER BY...  
  
    • String matching with like and regexp  
        • like -- whole string matching  
            • wildcard: % means zero or more characters  
            • wildcard: \_ means zero or one character  
        • regexp -- substring matching  
            • wildcard: .\*  
            • start of string anchor: ^  
            • end of string anchor: $  
    • Cross Product and Joins  
        • cross product -- pairs every row from table A with every row from table B  
         
                FROM A, B     
                 
        • join -- pairs ONLY linked rows from A and B  
             
                FROM A JOIN B ...  
                 
                • USING(field) -- EQUALITY between two FIELDS WITH THE SAME NAME  
             
                    FROM A JOIN B USING(id)  
                 
                • ON A.fieldname = B.fieldname -- Boolean comparison (True or False) between two fields (which can have different names)  
                 
                    FROM A JOIN B ON a.fieldname = b.fieldname  
             
        • Alias -- simplifies writing of SELECT statements, used when specifying fields  
     
                    FROM professors as p JOIN ...  
                    or  
                    FROM professors p JOIN ...

### Continued: Inserting and retrieving data, notes on joins.2.4.20.pdf 2/6/20

See Answers stored on previous Lecture.

### More SQL Queries, Multiple Tables 2/11/20

Attached Files:

* + [[File](https://learn.bu.edu/bbcswebdav/pid-7712399-dt-content-rid-36924993_1/xid-36924993_1) More SQL queries.2.6.20.pdf](https://learn.bu.edu/bbcswebdav/pid-7712399-dt-content-rid-36924993_1/xid-36924993_1) (47.152 KB)
  + [[File](https://learn.bu.edu/bbcswebdav/pid-7712399-dt-content-rid-37145777_1/xid-37145777_1) Answers More SQL queries.2.6.20.pdf](https://learn.bu.edu/bbcswebdav/pid-7712399-dt-content-rid-37145777_1/xid-37145777_1) (136.296 KB)

Uses Editor's Database.

### Continued: More SQL queries.2.6.20.pdf 2/13/20

Attached Files:

* + [[File](https://learn.bu.edu/bbcswebdav/pid-7734166-dt-content-rid-37349557_1/xid-37349557_1) Moving data between tables.2.13.20.pdf](https://learn.bu.edu/bbcswebdav/pid-7734166-dt-content-rid-37349557_1/xid-37349557_1) (39.432 KB)

See Answers stored on previous Lecture.

Topics:  
• Moving data into a new table using  Insert ... Select ...

**Union, Intersection, Set Difference 2/20/20**

Attached Files:

* [[File](https://learn.bu.edu/bbcswebdav/pid-7726897-dt-content-rid-37174736_1/xid-37174736_1) union intersection set difference outer joins 2.13.20.pdf](https://learn.bu.edu/bbcswebdav/pid-7726897-dt-content-rid-37174736_1/xid-37174736_1) (80.243 KB)
* [[File](https://learn.bu.edu/bbcswebdav/pid-7726897-dt-content-rid-37532881_1/xid-37532881_1) Answers union intersection set difference outer joins 2.13.20.pdf](https://learn.bu.edu/bbcswebdav/pid-7726897-dt-content-rid-37532881_1/xid-37532881_1) (148.604 KB)

Topics:  
  
• Refresher on joins  
    • Natural Join  
• Outer Joins  
    • Left  
    • Right  
• Union  
    • NO duplicates  
     
            SELECT ...  
            UNION  
            SELECT ...  
     
    • WITH Duplicates  
     
            SELECT...  
            UNION ALL  
            SELECT...  
     
• Intersection  
• Set Difference  
    • uses Outer Join

### Continued: union intersection set difference outer joins 2.13.20.pdf 2/25/20

See answers in previous lecture.

Introduction to CGI communication between a laptop browser and a server.

**Using CGI, Forms, HTML, MYSQL 2/27/20**

Attached Files:

* [[File](https://learn.bu.edu/bbcswebdav/pid-7742310-dt-content-rid-37504508_1/xid-37504508_1) test\_dbaccess\_cgi.py](https://learn.bu.edu/bbcswebdav/pid-7742310-dt-content-rid-37504508_1/xid-37504508_1) (2.558 KB)
* [[File](https://learn.bu.edu/bbcswebdav/pid-7742310-dt-content-rid-37504811_1/xid-37504811_1) where does my cgi program go?.pdf](https://learn.bu.edu/bbcswebdav/pid-7742310-dt-content-rid-37504811_1/xid-37504811_1) (51.018 KB)
* [[File](https://learn.bu.edu/bbcswebdav/pid-7742310-dt-content-rid-37532870_1/xid-37532870_1) test\_form\_cgi.py](https://learn.bu.edu/bbcswebdav/pid-7742310-dt-content-rid-37532870_1/xid-37532870_1) (1.993 KB)

Topics

Finish Union, Intersection, Set Difference problems

Using CGI, Forms, HTML, MYSQL

**Aggregate Functions 3/3/20**

Attached Files:

* [[File](https://learn.bu.edu/bbcswebdav/pid-7743544-dt-content-rid-37532873_1/xid-37532873_1) aggregate functions in select.2.21.20.pdf](https://learn.bu.edu/bbcswebdav/pid-7743544-dt-content-rid-37532873_1/xid-37532873_1) (46.783 KB)
* [[File](https://learn.bu.edu/bbcswebdav/pid-7743544-dt-content-rid-37709624_1/xid-37709624_1) Answers aggregate functions in select.2.21.20.pdf](https://learn.bu.edu/bbcswebdav/pid-7743544-dt-content-rid-37709624_1/xid-37709624_1) (142.179 KB)

**Nested Queries 3/5/20**

Attached Files:

* [[File](https://learn.bu.edu/bbcswebdav/pid-7751658-dt-content-rid-37709459_1/xid-37709459_1) Nested SQL Queries 3.5.20.pdf](https://learn.bu.edu/bbcswebdav/pid-7751658-dt-content-rid-37709459_1/xid-37709459_1) (36.64 KB)
* [[File](https://learn.bu.edu/bbcswebdav/pid-7751658-dt-content-rid-38059857_1/xid-38059857_1) Answers Nested SQL Queries 3.5.20.pdf](https://learn.bu.edu/bbcswebdav/pid-7751658-dt-content-rid-38059857_1/xid-38059857_1) (123.142 KB)

Topics:  
  
Subqueries  
• ANY  
    • [WHERE|HAVING] field operation ANY (SELECT...)  
  
                            Example:  
                             WHERE budget >= ANY (SELECT ...)  
  
• ALL  
    • WHERE|HAVING] field operation ALL (SELECT ...)  
  
                            Example:  
                             WHERE budget >= ALL (SELECT ...)  
  
• EXISTS  
    • [WHERE|HAVING] [EXISTS|NOT EXISTS] (SELECT...)

**Continued: Nested SQL Queries 3.5.20.pdf 3/17/20**

Attached Files:

* [[File](https://learn.bu.edu/bbcswebdav/pid-7776450-dt-content-rid-37929336_1/xid-37929336_1) Division in SQL 3.16.20.pdf](https://learn.bu.edu/bbcswebdav/pid-7776450-dt-content-rid-37929336_1/xid-37929336_1) (69.914 KB)

See Answers stored on previous Lecture.

Topics:  
  
• Division problem (find x in all y)  
    • Venn diagram to illustrate  
    • use WHERE NOT EXISTS to test for empty set  
    • correlated subquery which contains a field from an outer query  
  
            Example:  professors who are appointed in all departments  
             
                SELECT p.pid, p.lname  
                FROM Professor p  
                WHERE NOT EXISTS (SELECT did  
                                                     FROM department  
                                                      WHERE did not in (SELECT did  
                                                                                        FROM appointment  
                                                                                        WHERE pid = p.pid))

**Data Storage and Indexes, Binary Search Tree and B-Tree 3/19/20**

Attached Files:

* [[File](https://learn.bu.edu/bbcswebdav/pid-7789707-dt-content-rid-37999076_1/xid-37999076_1) Binary trees, B-trees, clustered and unclustered indexes.pdf](https://learn.bu.edu/bbcswebdav/pid-7789707-dt-content-rid-37999076_1/xid-37999076_1) (3.708 MB)
* [[File](https://learn.bu.edu/bbcswebdav/pid-7789707-dt-content-rid-37999077_1/xid-37999077_1) binary search tree.pdf](https://learn.bu.edu/bbcswebdav/pid-7789707-dt-content-rid-37999077_1/xid-37999077_1) (1.385 MB)

Topics:

* Data storage on flash memory (formerly hard disk, tape drive)  
  + Pages -- 4K bytes/page
  + Each page holds table records
* Basic table organization
  + Heap File
    - linear linking of consecutive pages
    - no index (quick way to look up records)
    - only operation is a scan which looks through each page, through each record
* Indexes
  + Basic idea is a search tree (binary search tree)
    - one test value per node
    - can find a value in log(n) tests, for n values
  + Index nodes are stored on flash memory, so retrieval of each node is slow
  + To speed up, add more test values to each node (and more pointers to lower nodes)
    - B+ Tree implements this
  + Number of pointers and number of pages determine depth of index tree
    - tree depth (number of layers with test values) = log\_p(N) where
      * p = number of pointers per tree node
      * N = number of data pages
* Clustered Index -- key k
  + B+ Tree index
  + k is a single field or a combination of fields (usually the "primary key")
  + **Data file is sorted by k**
* Unclustered index -- key uk
  + B+ Tree index
  + uk is a single field or a combination of fields
  + Index tree **contains an extra layer of pages at the bottom**
    - values for uk are sorted in the extra layer
    - each uk value is paired with a pointer to an individual data record in the data file
  + **Data file is NOT sorted by uk**

**Creating and Using Indexes 3/24/20**

Attached Files:

* [[File](https://learn.bu.edu/bbcswebdav/pid-7800723-dt-content-rid-38345418_1/xid-38345418_1) Creating and Using Indexes, Using Explain.ipynb](https://learn.bu.edu/bbcswebdav/pid-7800723-dt-content-rid-38345418_1/xid-38345418_1) (34.77 KB)

### Creating and Using Indexes continued 3/26/20