ENGR 3700U
Data Management Systems

Project Phase II

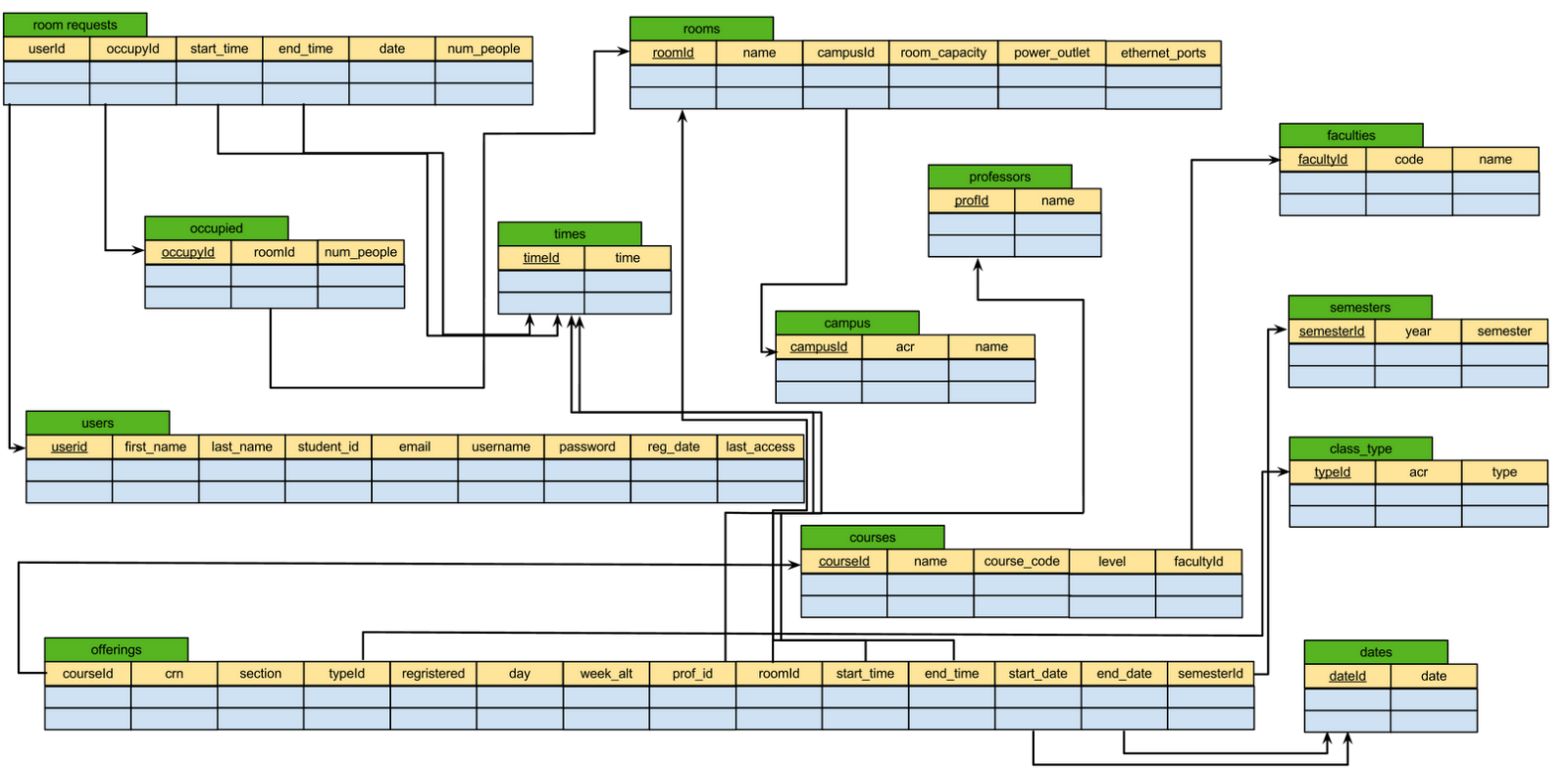
Jonathan Gillett (100437638)
Joseph Heron (100425488)
Amit Jain (100449233)
Wesley Unwin (100370215)
Minou Jhin (100434662)

Part A: Relational Schema

SQL Create Table Commands

Please see the included file *create_schema.sql*, which contains a complete list of all the SQL CREATE TABLE commands used to create all of the relations in our database. The script is imported in MySQL to create all of the relations for our database.

Graphical Diagram of Relations



Part B: Sample Data

Please see the included file *free_room_finder.db*, which contains a dump of our entire database, including the data entered in each table. The database dump can be imported in MySQL to create our entire database populated with data.

Part C: Views

View 1: SHOW_ALL_COURSES

Description of View

Shows all the courses that are offered in all of the years with relevant information and ordered by the course name.

SQL Create View Statement

```
CREATE VIEW SHOW_ALL_COURSES AS
SELECT
      c.name AS course_name,
      p.name AS professor_name,
      f.name AS faculty_name,
      f.code AS course_code,
      c.course_code,
      o.crn,
      t.time AS start_time,
      t2.time AS end_time,
      ct.type AS class_type,
      r.name,
      s.year,
      s.semester,
      o.section
FROM
      professors AS p,
      faculties AS f,
      courses AS c,
      offerings AS o,
      rooms AS r,
      times AS t,
      times AS t2,
      dates AS d,
      dates AS d2,
      class_type AS ct,
      semesters AS s
WHERE
```

```
      p.profId = o.profId AND
      c.courseId = o.courseId AND
      f.facultyId = c.facultyId AND
      r.roomId = o.roomId AND
      o.start_time = t.timeId AND
      o.end_time = t2.timeId AND
      o.start_date = d.dateId AND
      o.end_date = d2.dateId AND
      o.typeId = ct.typeId AND
      o.semesterId = s.semesterId
ORDER BY
      c.name;
```

Output

Note, the following is the result of one query, broken up into several sections in order to display all of it

```
+-----------------------------+----------------+-------------+-------------+
| course_name                 | professor_name | faculty_name | course_code |
+-----------------------------+----------------+-------------+-------------+
| Calculus I                  | Mihai Beligan  | Mathematics  | MATH        |
| Calculus II                 | Walid Ibrahim  | Mathematics  | MATH        |
| Data Management Systems     | Eyhab Al-Masri | Engineering  | ENGR        |
| Data Management Systems     | Eyhab Al-Masri | Engineering  | ENGR        |
| Discrete Mathematics        | Mihai Beligan  | Engineering  | ENGR        |
| Discrete Mathematics        | Walid Ibrahim  | Engineering  | ENGR        |
| Stats & Probability for Eng | Paula Dicato   | Statistics   | STAT        |
| Stats & Probability for Eng | Mihai Beligan  | Statistics   | STAT        |
+-----------------------------+----------------+-------------+-------------+

-------------+-------+------------+----------+------------+--------+------+
 course_code | crn   | start_time | end_time | class_type | name   | year |
-------------+-------+------------+----------+------------+--------+------+
 1010        | 40288 | 12:10:00   | 13:30:00 | Lecture    | UB2080 | 2012 |
 1020        | 41058 | 08:10:00   | 09:30:00 | Lab        | UA1350 | 2012 |
 3700        | 42578 | 18:40:00   | 20:00:00 | Lecture    | UA1240 | 2012 |
 3700        | 42578 | 17:10:00   | 18:30:00 | Lecture    | UA1240 | 2012 |
 2110        | 41063 | 08:10:00   | 09:30:00 | Lecture    | UA1240 | 2012 |
 2110        | 41064 | 13:40:00   | 15:00:00 | Lab        | UA2230 | 2012 |
 2800        | 70247 | 08:10:00   | 09:30:00 | Lecture    | UA1350 | 2012 |
 2800        | 70248 | 10:40:00   | 12:00:00 | Tutorial   | UA2230 | 2012 |
-------------+-------+------------+----------+------------+--------+------+

----------+---------+
 semester | section |
----------+---------+
 Fall     | 001     |
 Fall     | 001     |
 Fall     | 001     |
 Fall     | 001     |
 Fall     | 001     |
 Fall     | 002     |
 Fall     | 001     |
 Fall     | 002     |
----------+---------+
```

View 2: COURSE_CODE_H_T

Description of View

List all the courses with the number of courses offered that contain an ‘H’ or ‘T’ in the course code.

SQL Create View Statement

```
CREATE VIEW COUSE_CODE_H_T AS
SELECT
    c.name,
    COUNT(o.courseId) AS number_of_courses
FROM
    offerings AS o,
    faculties AS f,
    professors AS p,
    courses AS c
WHERE
    p.profId = o.profId AND
    f.facultyId = c.facultyId AND
    c.courseId = o.courseId AND
    f.code = ANY
    (SELECT
           f2.code
     FROM
           faculties AS f2
     WHERE
           f2.code LIKE '%H%' OR
           f2.code LIKE '%T%')
GROUP BY
        c.name;
```

Output

```
+----------------------------+--------------------+
| name                       | number_of_courses |
+----------------------------+--------------------+
| Calculus I                 |                  1 |
| Calculus II                |                  2 |
| Stats & Probability for Eng |                  2 |
+----------------------------+--------------------+
```

View 3: COURSES_AT_TIME

Description of View

Returns all course names that start at the specified time (where that time is whether any two courses from the same faculty share the same start time regardless of day).

SQL Create View Statement

```
CREATE VIEW COURSES_AT_TIME AS
SELECT
    c.name
FROM
    courses AS c,
    offerings AS o,
    faculties AS f
WHERE
    c.facultyId = f.facultyId AND
    c.courseId = o.courseId AND
    o.start_time = ANY
    (SELECT
        o2.start_time
    FROM
        offerings AS o2,
        courses AS c2,
        faculties AS f2
    WHERE
        c2.facultyId = f2.facultyId AND
        c2.courseId = o2.courseId AND
        f2.facultyId <> f.facultyId);
```

Output

```
+-----------------------------+
| name                        |
+-----------------------------+
| Stats & Probability for Eng |
| Discrete Mathematics        |
| Calculus II                 |
+-----------------------------+
```

View 4: ALL_CLASSES_ROOMS

Description of View

Display all of the rooms on campus with the day and room capacity. **Please note that since MySQL does not support full outer joins this query will not work.**[1]

SQL Create View Statement

```
CREATE VIEW ALL_CLASSES_ROOMS AS
SELECT
    o.courseId AS course_name,
    o.day,
    r.name AS room_name,
    r.room_capacity
FROM
    rooms AS r FULL JOIN
    offerings AS o ON
    r.roomId = o.roomId;
```

SQL Create View Statement MySQL compliant

```
CREATE VIEW ALL_CLASSES_ROOMS AS
(SELECT
    o.courseId AS course_name,
    o.day,
    r.name AS room_name,
    r.room_capacity
FROM
    rooms AS r RIGHT JOIN
    offerings AS o ON
    r.roomId = o.roomId)
UNION
(SELECT
    o.courseId AS course_name,
    o.day,
    r.name AS room_name,
    r.room_capacity
```

[1] https://dev.mysql.com/doc/refman/5.6/en/join.html

```
FROM
    rooms AS r LEFT JOIN
    offerings AS o ON
    r.roomId = o.roomId);
```

Please note: That although the first view is a valid SQL query, **full joins are not supported in MySQL**, a simple full join can still be mimicked with a ‘Left Join’ ‘Union’ a ‘Right Join’ given by the MySQL compliant command.

Output

```
+-------------+------+-----------+---------------+
| course_name | day  | room_name | room_capacity |
+-------------+------+-----------+---------------+
|           1 | T    | UA1350    |           250 |
|           1 | R    | UA2230    |            50 |
|           2 | M    | UA1240    |            75 |
|           2 | R    | UA1240    |            75 |
|           3 | M    | UB2080    |           175 |
|           4 | T    | UA1240    |            75 |
|           4 | T    | UA2230    |            50 |
|           5 | M    | UA1350    |           250 |
|           5 | W    | NULL      |          NULL |
|        NULL | NULL | UB2050    |            50 |
|        NULL | NULL | UA1220    |           150 |
+-------------+------+-----------+---------------+
```

View 5: TEACHING_MATH_ENGR

Description of View

Get all the professors that are teaching courses for the MATH faculty or the ENGR faculty.

SQL Create View Statement

```
CREATE VIEW TEACHING_MATH_ENGR AS
(SELECT
      p.name
FROM
      professors AS p,
      offerings AS o,
      courses AS c,
      faculties AS f
WHERE
      p.profId = o.profId AND
      c.courseId = o.courseId AND
      f.facultyId = c.facultyId AND
      f.code LIKE 'MATH')
UNION
(SELECT
      p1.name
FROM
      professors AS p1,
      offerings AS o1,
      courses AS c1,
      faculties AS f1
WHERE
      p1.profId = o1.profId AND
      c1.courseId = o1.courseId AND
      f1.facultyId = c1.facultyId AND
      f1.code LIKE 'ENGR');
```

Output

```
+----------------+
| name           |
+----------------+
| Mihai Beligan  |
| Walid Ibrahim  |
| Kamran Sartipi |
| Eyhab Al-Masri |
+----------------+
```

View 6: ROOMS_MOST_USED

Description of View

Show the the room name and times (per interval) that the room is taken, ordered by the rooms that have the most time slots and are therefore the most used.

SQL Create View Statement

```
CREATE VIEW ROOMS_MOST_USED AS
SELECT
      r.name AS room_name,
      TIMEDIFF(t2.time, t.time) AS duration
FROM
      offerings AS o,
      rooms AS r,
      times AS t,
      times AS t2,
      dates AS d,
      dates AS d2,
      class_type AS ct,
      semesters AS s
WHERE
      r.roomId = o.roomId AND
      o.start_time = t.timeId AND
      o.end_time = t2.timeId AND
      o.start_date = d.dateId AND
      o.end_date = d2.dateId AND
      o.typeId = ct.typeId AND
      s.semesterId = o.semesterId
GROUP BY
      r.roomId, o.day
ORDER BY
      duration DESC;
```

Output

```
+-----------+----------+
| room_name | duration |
+-----------+----------+
| UA2230    | 01:20:00 |
| UA1350    | 01:20:00 |
| UA1350    | 01:20:00 |
| UA1240    | 01:20:00 |
| UA1240    | 01:20:00 |
| UB2080    | 01:20:00 |
| UA1240    | 01:20:00 |
+-----------+----------+
```

View 7: ROOMS_UA

Description of View

This will return the rooms used on the North Oshawa Campus for the UA building.

SQL Create View Statement

```
CREATE VIEW ROOMS_UA AS
SELECT
       r.name, o.day, t.time AS start_time, t2.time AS end_time
FROM
      faculties AS f,
      courses AS c,
      offerings AS o,
      rooms AS r,
      times AS t,
      times AS t2,
      dates AS d,
      dates AS d2,
      class_type AS ct,
      campus AS ca,
      semesters AS s
WHERE
      c.courseId = o.courseId AND
      f.facultyId = c.facultyId AND
      r.roomId = o.roomId AND
      o.start_time = t.timeId AND
      o.end_time = t2.timeId AND
      o.start_date = d.dateId AND
      o.end_date = d2.dateId AND
      o.typeId = ct.typeId AND
      o.semesterId = s.semesterId AND
      s.year = YEAR(CURDATE()) AND
      ca.campusId = r.campusId AND
      ca.acr = 'UON' AND
      r.name LIKE '%UA%';
```

Output

```
+--------+-----+------------+----------+
| name   | day | start_time | end_time |
+--------+-----+------------+----------+
| UA1350 | T   | 08:10:00   | 09:30:00 |
| UA2230 | R   | 10:40:00   | 12:00:00 |
| UA1240 | M   | 18:40:00   | 20:00:00 |
| UA1240 | R   | 17:10:00   | 18:30:00 |
| UA1240 | T   | 08:10:00   | 09:30:00 |
| UA2230 | T   | 13:40:00   | 15:00:00 |
| UA1350 | M   | 08:10:00   | 09:30:00 |
+--------+-----+------------+----------+
```

View 8: USER_AND_REQUESTS

Description of View

Display all users and their corresponding room requests.

SQL Create View Statement

```
CREATE VIEW USERS_AND_REQUESTS AS
SELECT users.userId,
       users.first_name,
       users.last_name,
       users.student_id,
       rooms.name AS room
FROM  users,
      room_requests,
      occupied,
      rooms
WHERE users.userId = room_requests.userId AND
      room_requests.occupyId = occupied.occupyId AND
      occupied.roomId = rooms.roomId;
```

Output

```
+--------+------------+-----------+------------+--------+
| userId | first_name | last_name | student_id | room   |
+--------+------------+-----------+------------+--------+
|      1 | joseph     | hero      | 1002312344 | UA2230 |
|      1 | joseph     | hero      | 1002312344 | UA1350 |
|      2 | billy      | weds      | 1002231344 | UA1350 |
|      3 | super      | hero      | 1002989344 | UA1220 |
|      3 | super      | hero      | 1002989344 | UA1240 |
|      4 | you        | nu        | 1002314323 | UB2050 |
|      4 | you        | nu        | 1002314323 | UB2080 |
|      5 | gnu        | user      | 1002312044 | UB2080 |
|      6 | tinfoil    | hat       | 1003213098 | UA1240 |
|      3 | super      | hero      | 1002989344 | UA1240 |
+--------+------------+-----------+------------+--------+
```

View 9: TOTAL_PEOPLE_ROOM

Description of View

Show the total number of people expected in each room, for a particular time.

SQL Create View Statement

```
CREATE VIEW TOTAL_PEOPLE_ROOM AS
SELECT
    r.name, oc.num_people AS total_people
FROM
    rooms AS r,
    times AS t,
    times AS t2,
    occupied AS oc,
    room_requests AS rr
WHERE
    r.roomId = oc.roomId AND
    oc.occupyId = rr.occupyId AND
    rr.start_time = t.timeId AND
    rr.end_time = t2.timeId
GROUP BY
    r.name
ORDER BY
      oc.num_people;
```

Output

```
+--------+--------------+
| name   | total_people |
+--------+--------------+
| UB2050 |            1 |
| UA1220 |           10 |
| UA2230 |           20 |
| UA1240 |           25 |
| UB2080 |           50 |
| UA1350 |          100 |
+--------+--------------+
```

View 10: CLASSES_FOR_COURSE

Description of View

Display all classes related to a particular course. This example shows the classes offered for the course ‘ENGR 3700’.

SQL Create View Statement

```
CREATE VIEW CLASSES_FOR_COURSE AS
SELECT faculties.code,
       courses.course_code,
       class_type.acr,
       offerings.section,
       semesters.semester,
       offerings.week_alt,
       offerings.day,
       rooms.name
FROM courses,
     offerings,
     faculties,
     class_type,
     semesters,
     rooms
WHERE courses.course_code = '3700'               AND
      semesters.semesterId = offerings.semesterId AND
      courses.facultyId   = faculties.facultyId  AND
      faculties.code      = 'ENGR'               AND
      courses.courseId    = offerings.courseId   AND
      offerings.roomId    = rooms.roomId         AND
      offerings.typeId    = class_type.typeId;
```

Output

```
+------+-------------+------+---------+----------+----------+-----+---------+
| code | course_code | acr  | section | semester | week_alt | day | name    |
+------+-------------+------+---------+----------+----------+-----+---------+
| ENGR | 3700        | LEC  | 001     | Fall     |     NULL | M   | UA1240 |
| ENGR | 3700        | LEC  | 001     | Fall     |     NULL | R   | UA1240 |
+------+-------------+------+---------+----------+----------+-----+---------+
```

Part D: Entity-Relationship Diagram