IS607 Week 05 Project

**Database Purpose**

I propose to build a database to support the after-school activities signup. This database will organize the after-school activities and keep track of the students and teachers per activity.

Five Tables and SQL Code

Table 1: Teacher Table

|  |  |
| --- | --- |
| Teacher\_id | Teacher\_name |
| 1 | Andy Catlin |
| 2 | Mike Schulte |
| 3 | Rokshana Ali |
| 4 | Nidia Cordova |

Table 2: Student Table

|  |  |
| --- | --- |
| Student\_id | Student\_name |
| 1 | Honey Berk |
| 2 | James Hamski |
| 3 | Raj Srinivasan |

Table 3: Activities Table

|  |  |  |
| --- | --- | --- |
| Activities\_id | Activities Description | Teacher\_id |
| 1 | Art | 3 |
| 2 | Cooking | 1 |
| 3 | Sports | 2 |

Table 4: Registration Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Registration | Student\_id | Activities\_id | State | Price |
| Spring 2015 | 1 | 2 | NY | 200 |
| Spring 2015 | 2 | 3 | NY | 200 |
| Spring 2015 | 3 | 1 | NY | 200 |

Table 5: Semester

|  |  |
| --- | --- |
| Semester\_id | Semester\_name |
| 1 | Fall 2014 |
| 2 | Spring 2015 |

**SQL Code:**

**#create tables**

DROP TABLE IF EXISTS tbl\_teacher;

DROP TABLE IF EXISTS tbl\_student;

DROP TABLE IF EXISTS tbl\_activities;

DROP TABLE IF EXISTS tbl\_registration;

CREATE TABLE tbl\_teacher (teacher\_id int PRIMARY KEY, teacher\_name varchar NOT NULL);

INSERT INTO tbl\_teacher (teacher\_id, teacher\_name) VALUES ( 1, 'Andy Caitlin');

INSERT INTO tbl\_teacher (teacher\_id, teacher\_name) VALUES ( 2, 'Mike Schulte');

INSERT INTO tbl\_teacher (teacher\_id, teacher\_name) VALUES ( 3, 'Rokshana Ali');

INSERT INTO tbl\_teacher (teacher\_id, teacher\_name) VALUES ( 4, 'Nidia Cordova');

SELECT \* FROM tbl\_teacher;

CREATE TABLE tbl\_student (student\_id int PRIMARY KEY, student\_name varchar NOT NULL);

INSERT INTO tbl\_student (student\_id, student\_name) VALUES ( 1, 'Honey Berk');

INSERT INTO tbl\_student (student\_id, student\_name) VALUES ( 2, 'James Hamski');

INSERT INTO tbl\_student (student\_id, student\_name) VALUES ( 3, 'Raj Srinivasan');

SELECT \* FROM tbl\_student;

CREATE TABLE tbl\_activities (activities\_id int PRIMARY KEY, activities\_description varchar NOT NULL, teacher\_id int);

INSERT INTO tbl\_activities (activities\_id, activities\_description,teacher\_id) VALUES ( 1, 'Art', 3);

INSERT INTO tbl\_activities (activities\_id, activities\_description,teacher\_id) VALUES ( 2, 'Cooking', 1);

INSERT INTO tbl\_activities (activities\_id, activities\_description,teacher\_id) VALUES ( 3, 'Sport', 2);

SELECT \* FROM tbl\_activities;

CREATE TABLE tbl\_registration (registration varchar NOT NULL, student\_id int, activities\_id int,

state varchar NOT NULL, price numeric, PRIMARY KEY (student\_id, activities\_id));

INSERT INTO tbl\_registration (registration, student\_id, activities\_id, state, price) VALUES ( 'Spring 2015',1, 2, 'NY', 200);

INSERT INTO tbl\_registration (registration, student\_id, activities\_id, state, price) VALUES ( 'Spring 2015',2, 3, 'NY', 200);

INSERT INTO tbl\_registration (registration, student\_id, activities\_id, state, price) VALUES ( 'Spring 2015',3, 1, 'NY', 200);

SELECT \* FROM tbl\_registration;

**#one to many example**

**#teacher with NULL activities does not show up**

SELECT

t.teacher\_id, t.teacher\_name, a.activities\_id, a.activities\_description

FROM tbl\_teacher t

JOIN tbl\_activities a

ON t.teacher\_id=a.teacher\_id

ORDER BY t.teacher\_id

**#one to many example**

**#teacher with NULL activities shows up**

SELECT

t.teacher\_id, t.teacher\_name, a.activities\_id, a.activities\_description

FROM tbl\_teacher t

LEFT JOIN tbl\_activities a

ON t.teacher\_id=a.teacher\_id

ORDER BY t.teacher\_id

**#one to many example**

**#replace null value with zero or 'NA'**

SELECT

t.teacher\_id, t.teacher\_name, COALESCE(a.activities\_id,0), COALESCE(a.activities\_description,'NA')

FROM tbl\_teacher t

LEFT JOIN tbl\_activities a

ON t.teacher\_id=a.teacher\_id

ORDER BY t.teacher\_id

**#many to many example**

SELECT \* FROM tbl\_registration r

INNER JOIN tbl\_activities a

ON r.activities\_id=a.activities\_id

INNER JOIN tbl\_teacher t

ON a.teacher\_id = t.teacher\_id

**#many to many example**

SELECT \* FROM tbl\_registration r

INNER JOIN tbl\_activities a

ON r.activities\_id=a.activities\_id

INNER JOIN tbl\_student s

ON r.student\_id=s.student\_id

INNER JOIN tbl\_teacher t

ON a.teacher\_id = t.teacher\_id

Short requirement on how to extend database functionality

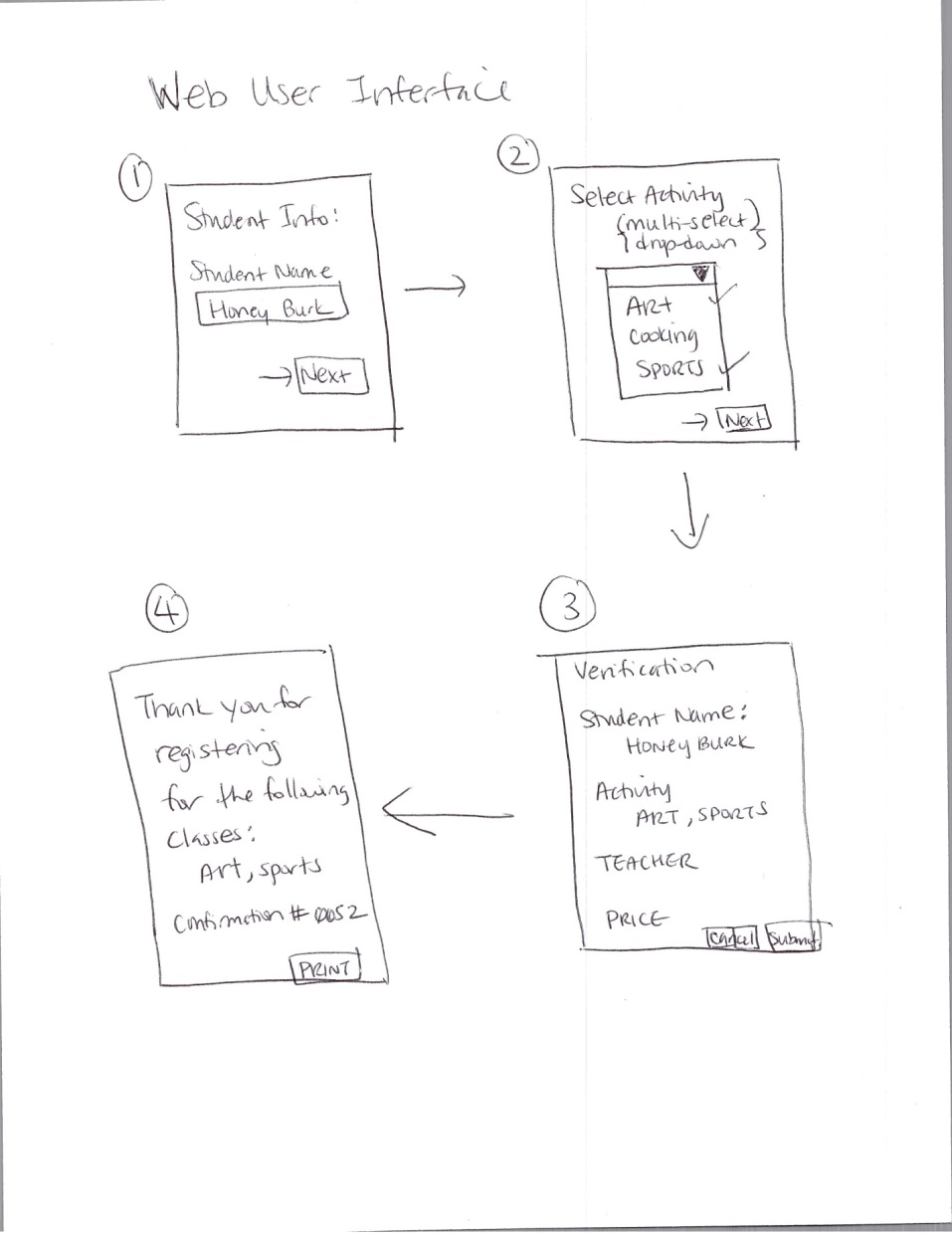
**Database Improvements**

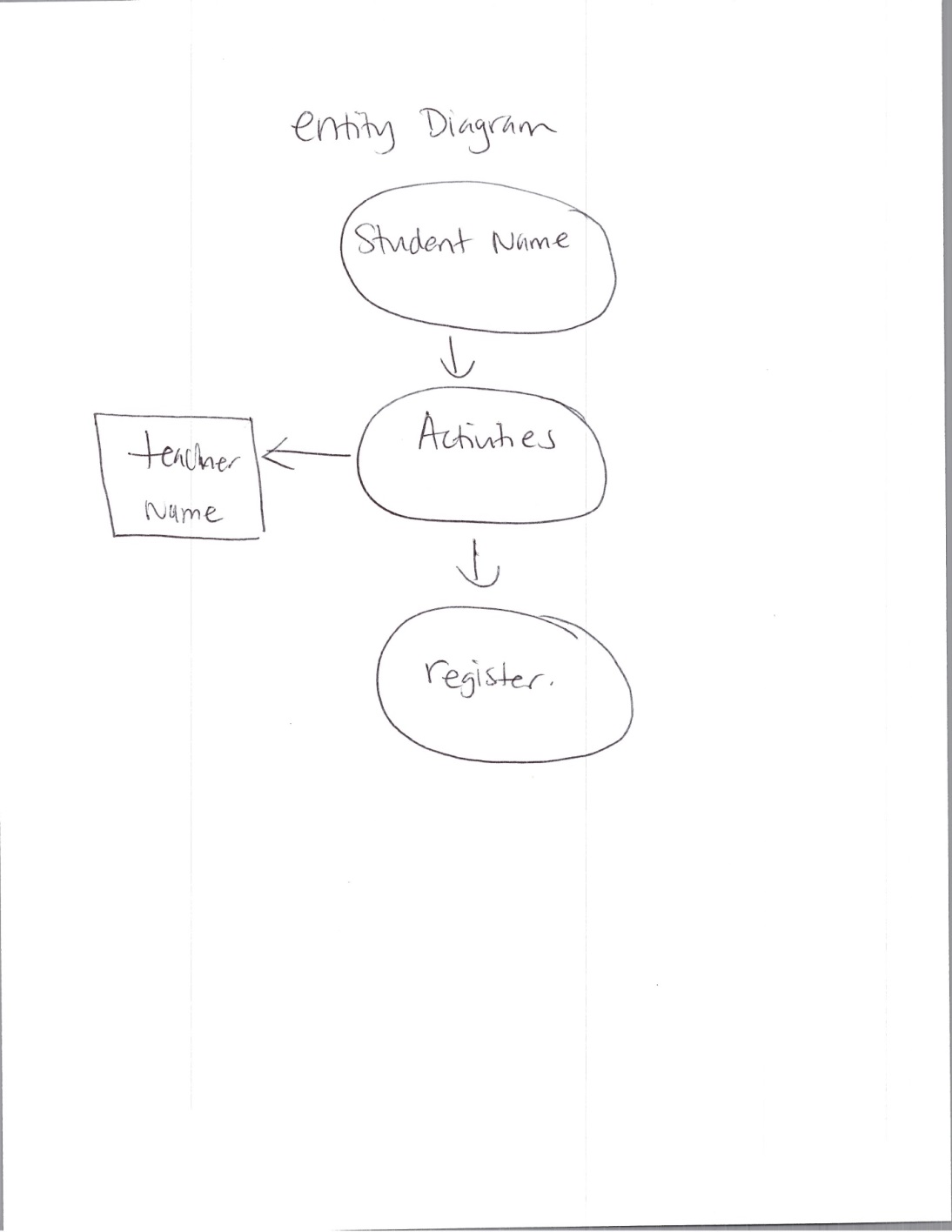
Adding a time variable will improve the database. For example, by adding activity start date and activity end date, we can track the students and teachers participation.

**Referential integrity**

This concept refers to any table with a foreign key relationship. The relationship between two tables that share a foreign key must remain consistent. To keep the integrity and be consistent, you cannot add or delete a record to a table that contains foreign keys unless you add or delete the record from the linked table. For example, in the activities table, I cannot add an activity without updating the teacher\_id in the teacher’s table. A technique called cascading update or delete can modify the table.

**Web User Interface & Entity Diagram**



****