**PROJECT PHASE I-SPECIFICATION**

1) **Team members**:

Ahmed Hassan M. H. Ibrahim   - 160709057

Hasibullah Mahmood                - 160709073

2) **Data explanation**:

. **The purpose of your project**:

To learn how to create databases, manage and query them. Also, to get our hands dirty with MySQL workbench and querying in SQL.

* **Number of existing files :**

     12 files

|  |  |  |
| --- | --- | --- |
| **Name of files** | **Number of rows** | **Number of columns** |
| address | 672 | 17 |
| charter\_type | 4 | 2 |
| community | 77 | 2 |
| geographic\_area | 30 | 2 |
| governance | 3 | 2 |
| grades | 1337 | 4 |
| NCES | 672 | 4 |
| program\_types | 33 | 2 |
| school\_category | 3 | 2 |
| school\_program\_types | 878 | 2 |
| school\_type | 11 | 4 |
| schools | 672 | 11 |

.      **Kaggle address of the raw data**:

<https://www.kaggle.com/chicago/chicago-cps-schools-2013-2014-academic-year>

**·       Bitbucket URL of the ER diagram (.mwb file)**

<https://bitbucket.org/U160709073/database_project/src/>

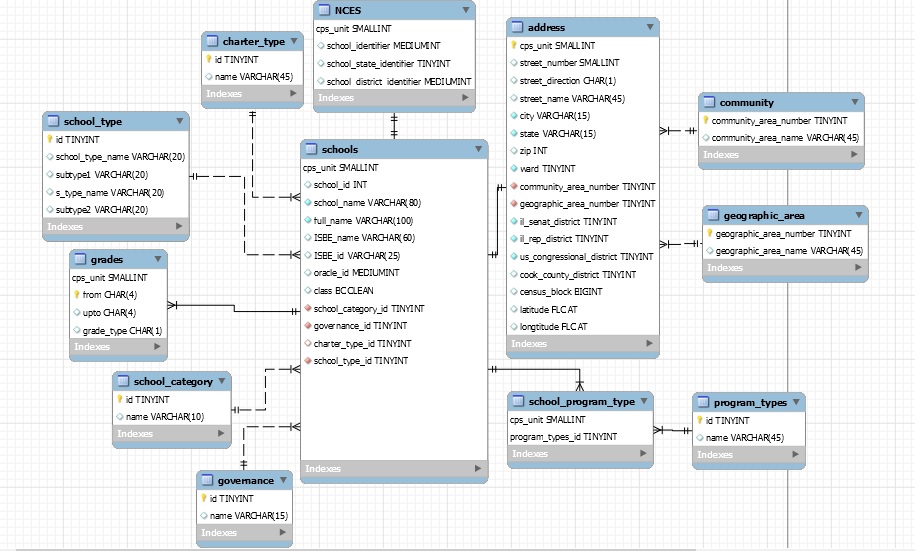
3) **10 English questions to query the database system:**

1. Show the schools full name which exist in “Bronzeville” Geographic

Area?

1. Show the ES schools in “Logan square” community area?
2. Show the HS schools which has a contract type?
3. Show the schools which have PK-8th attending grades and Early\_Childhood\_Program in 2th US Congressional District?
4. Show the community area that has the most schools in it?
5. Show the community area that has the least schools in it?
6. Show the average number of schools according to geographic area?
7. Show all the ES schools, community area west town and located in ward 1?
8. Which school has the most programs?
9. What is the popular program in schools?

**4) Relational data model (ER diagram):**



**5) The SQL statements that will implement the English questions:**

1. SELECT full\_name

From schools

WHERE cps\_unit in (SELECT cps\_unit

FROM address JOIN geographic\_area USING(geographic\_area\_number)

WHERE geographic\_area\_name = "Bronzeville");

    2. SELECT full\_name

FROM schools JOIN school\_category

WHERE schools.school\_category\_id = school\_category.id

AND school\_category.name = "ES" AND cps\_unit IN (SELECT cps\_unit

FROM address JOIN community USING(community\_area\_number)

WHERE community\_area\_name = "Logan square");

    3. SELECT full\_name

FROM schools JOIN school\_category

JOIN school\_type

WHERE schools.school\_category\_id = school\_category.id

AND schools.school\_type\_id = school\_type.id

AND school\_category.name = "HS"

AND school\_type\_name = "contract";

      4. SELECT full\_name

FROM schools JOIN grades USING(cps\_unit)

WHERE grades.from = "PK"

AND grades.upto = "8th"

AND cps\_unit IN (SELECT cps\_unit

FROM school\_program\_type JOIN schools USING(cps\_unit)

JOIN program\_types

WHERE school\_program\_type.program\_types\_id = program\_types.id

AND program\_types.name = "Early\_Childhood\_Program"

AND cps\_unit IN (SELECT cps\_unit

FROM schools JOIN address USING(cps\_unit)

WHERE us\_congressional\_district = 2));

5. SELECT community\_area\_name, COUNT(\*) AS count

FROM address JOIN community USING(community\_area\_number)

GROUP BY community\_area\_name

ORDER BY count DESC

LIMIT 1;

   6. SELECT community\_area\_name, COUNT(\*) AS count

FROM address JOIN community USING(community\_area\_number)

GROUP BY community\_area\_name

ORDER BY count

LIMIT 3;

   7. SELECT ROUND(AVG(counts)) AS average

FROM (

SELECT COUNT(\*) AS counts

FROM address JOIN geographic\_area USING(geographic\_area\_number)

GROUP BY geographic\_area\_name) AS counts;

   8. SELECT full\_name

FROM schools JOIN school\_category

ON schools.school\_category\_id = school\_category.id

WHERE school\_category.name = "ES"

AND cps\_unit IN (SELECT cps\_unit

FROM address JOIN schools USING(cps\_unit)

JOIN community USING(community\_area\_number)

WHERE community\_area\_name = "west town"

AND ward = 1);

   9. SELECT full\_name

FROM schools JOIN

(

SELECT cps\_unit, count(cps\_unit) counts

FROM schools JOIN school\_program\_type USING(cps\_unit)

ORDER BY counts DESC

LIMIT 1) AS NEWTABLE USING(cps\_unit);

10.   SELECT \*

FROM program\_types JOIN

(SELECT program\_types\_id AS id, count(program\_types\_id) counts

FROM school\_program\_type JOIN schools USING(cps\_unit)

GROUP BY program\_types\_id) AS program\_type\_count USING(id)

ORDER BY counts DESC;

**VIEW:**

CREATE VIEW schools\_per\_geogrphic\_area AS

SELECT geographic\_area\_name, COUNT(\*) AS counts

FROM address JOIN geographic\_area USING(geographic\_area\_number)

GROUP BY geographic\_area\_name

ORDER BY counts DESC;

**STORED PROCEDURES:**

CREATE DEFINER=`root`@`localhost` PROCEDURE `SchoolsForGivenCategory`(IN category CHAR(2))

BEGIN

SELECT full\_name

FROM schools JOIN school\_category ON school\_category\_id = id

WHERE name = category

ORDER BY full\_name;

END

CALL SchoolsForGivenCategory('MS');

--------------------------------------------------------------------------------------------------------------------------------------------

CREATE DEFINER=`root`@`localhost` PROCEDURE `GovernanceForGivenCPSunit `(IN cps SMALLINT, OUT governance\_type VARCHAR(15))

BEGIN

SELECT name INTO governance\_type

FROM schools JOIN governance ON governance\_id = id

WHERE cps\_unit=cps;

END

CALL GovernanceForGivenCPSunit(1105, @governanceType);

SELECT @governanceType;

CREATE DEFINER=`root`@`localhost` PROCEDURE ` school\_with\_most\_programs`(OUT schoolName VARCHAR(100))

BEGIN

SELECT full\_name INTO schoolName

FROM schools JOIN

(

SELECT cps\_unit, count(cps\_unit) counts

FROM schools JOIN school\_program\_type USING(cps\_unit)

ORDER BY counts DESC

LIMIT 1) AS NEWTABLE USING(cps\_unit);

END

CALL school\_with\_most\_programs(@schoolName);

SELECT @schoolName;

**6) The way of loading the database with values:**

We downloaded the data from Gaggle website. Then, we cleaned the data by coding some python scripts. After learning the data we created the ER diagram. Then, we created .CSV file for each entity. We will add the data by using MySQL workbench “import records from external file” in the inserts choice for each entity.

**7) GUI:**

Creating a map which will take each longitude and latitude of every school and make a pointer on the place of each school then when the user click on the pointer it will show to them a summary of each school. An enabling GPS location to measure the distance between the user and specific school.

**8) The database platform we plan to use:**

MySQL 8.0 ON PC LAPTOP INTEL CORE i5, RAM 4 GB, 500 HDD