School Management System

PROJECT REPORT

Group Members

1. SENG CHETRA
2. PUN MENGLY
3. PHAT CHETRA
4. TRY SORTA

Acknowledgement

I would like give my special thanks to my database teacher, Mr. Sovisal Chenda for giving us this project. Thank you for all the great helps you offer by reviewing the documents and the advice you gave me was really helpful. I wouldn’t be able to achieve this without your help and support.

Big thanks to Pun Mengly for his great API implementation making this project possible. We couldn’t have done it without you. Thank you for hard work and participation. I also want to give my full appreciation to other members such as Phat Chetra and Try Sorta for participating in this wonderful project.

Contents

I. PROPOSAL

Change is inevitable and same goes with how we live and learn. We rely so much on ourselves and other people that we overlook how much technologies can make our lives specifically students' lives better. New technologies are game changers. That's why most universities are using it. Modern technologies and software are being implemented into their systems every day while our school still use the old traditional learning approach for students and teachers.

1.1 PROBLEMS

Royal University of Phnom Penh is far behind the competition in term of the technologies that are being used. In fact, we don't even have a database that enables students to view their grade in previous semesters, or one that allows teacher to see how many students there are in a particular class. It's time to change and revolutionize our learning approach by implement this so called "New technology" into our school.

1.2 SOLUTIONS

To response to this issue, our team decided to build and implement our own school database, which will drastically improve our school system and at the same time, becomes beneficial for everyone including students and lecturers. Our team's project is cost-efficient meaning that our school doesn't need to hire expertise to get the job done. Therefore, it's a win-win situation, since our team can gain experience from completing the project, and we can save thousands of dollars from building the database as well.

1.2.1 Our approach

We've tried every possible way to create a good and reliable database. Because we have the vision to change and have a clear understand of what problem we have to deal with, our team decided that the tables listed below will be in our database.

- students: This table will include all necessary data about students such as id, name, birthday, phone and so on.
- grades: A table that stores student's grade on each subject
- courses: Course's id, duration, description and other related information goes here
- teachers: This is where we store teachers' information including courses' id
- subjects: This table includes information about subject's maximum scores, lecturers, credits and so on

PS: Some tables might be added or modify in our final database.

1.2.2 Requirement

As mentioned above, our project will definitely be cost-efficient. However, that doesn't mean our team don't need resources to bring our database into reality. These are two things that have to be put into consideration, which are data, time, and a lot of data.

First, we'll be starting with a few data to demonstrate the capability of our database. Next, comes optimization. With enough data and time, we'll be able to optimize our code to be faster and more memory-efficient. Making sure that our system can run own minimum memory requirement will be our priority. After that, we'll be doing a few tests with our database and make some adjustments before completing our database system. Finally, we'll implement an API built specifically for our database.

In order to ensure a success in implementing our own database for the school, we've developed a manageable timeline as shown below:

- March 7th – March 20th 2021: System Analysis
- March 21st – April 3rd 2021: Data implementation
- April 4th – April 14th 2021: Project Finalization
- April 15th – April 21st 2021: Project final report

1.3 BENEFITS

Our team's database will change the way students and teachers in our school interact with technologies, and we're sure they will love it as much as we do. Our system will definitely be modernized and most importantly, it also gives us a step up the game and to join the competition again. Almost every university has made database and API for their students to used, and it's about time we make ours. Additionally, the use of database and API will provide other educational benefits for our students such as:

- Providing a better and modern learning approach by giving the opportunity for students to use database system
- Improving our school standard by implementing new technologies like databases compared to other famous universities
- Providing convenience and improve accessibility meaning that authorized users can get data at any time and any where
- Providing convenience and improve accessibility meaning that authorized users can get data at any time and any where
- Encourage students to improve and extend our database functionality in future, which in turn give them experience and further increase their knowledge of database system.

1.4 SUMMARY

In summary, our school is far behind the competition. We need some minor improvements on what technologies that are being used and how they can be used. Not having our own database for accessibility and availability is problematic, but our team has found the most suitable solution, which is also cost-efficient. With the given time and minimum resources and investment, it can deliver beneficial impacts on the whole school.

Nothing stays the same, everything changes and we strongly believe that the minor changes we make every day by implement new technologies can dramatically change our school's reputation and the overall learning quality for the students.

II. SYSTEM ANALYSIS

Behind every successful system, there's always a good design and planning. Before we can make a reliable and stable database, System analysis is required. That's why, our team took this approach and put our best efforts into designing a simplified, yet reliable database. We took our time and made our own system analysis to help us virtualize how our database would look like, understand what attributes should be included in each table, study how tables are connected, and we've also gathered some data to test when implementing our database.

2.1 TABLES

Having a lot of data doesn't mean our database is ready to use. We need ways to construct and categorize how those data should be organized. Our database's main purpose is to help managing students' grades in our school. Therefore, any information relating students' grade such are teachers, courses, grades are mandatory. We've tested many ways to simplified how our students' data should be categorize inside specific tables.

1. Mandatory tables

To successfully structure and organize the data, we need to understand the main focus and the fundamental tables inside our database. As previously mentioned, information about a particular student is essential for our database. That's why the tables listed below will definitely be included:

- Students table: this table should all information about a specific student
- Grades table: the most important table and sole purpose of our database creation
- Teachers table: schools need teachers, so this table is required.
- Courses table: this table will show the courses that students have enrolled for
- Subjects: every course has subjects, so this table is required.

2. Other supported tables

Although the tables listed below are crucial, some tables must be not be overlooked. Some tables might not be our primary focus in the system, but they exist to support our database. Without them, the database would not be able to function properly, and these tables include:

- Enrollments table: a link between students and courses tables
- Instructors table: a link between subjects and teachers tables
- Attendances table: this table should also be included.
- Parents table: this table should include students' parents contact

2.2 METADATA

To make sure our system is reliable, our database should be able store as many crucial information as we can, that relates to students. In order to achieve that, our team has decided that each table will have the columns as list below:

1. **students table**
- stu_id: student's ID, datatype INT and can't be null
- stu_name: student's name, datatype VARCHAR(25) and can't be null
- stu_gender: student's gender, datatype CHAR(1) and can be null
- stu_bd: student's birthday, datatype DATE and can't be null
- stu_phone: student's phone number, datatype VARCHAR(10) and can't be null
- stu_address: student's address, datatype VARCHAR(100)
- stu_class: student's class, datatype VARCHAR(4) and can't be null
- stu_year: student's academic year, datatype SMALLINT and can't be null
- stu_status: datatype BIT, 1 if the student is active in school and can't be null
- parent_id: datatype INT and will be used in database's relationship

2. **attendances table**
- att_id: student's attendance ID, datatype INT and can't be null
- att_date: attendance's date, datatype DATETIME and can't be null
- att_status: datatype BIT, 1 if attended and 0 if absent, can't be null
- att_remark: datatype VARCHAR(25), attendance's note such as late or permission
- stu_id: datatype INT and will be used in database's relationship

3. **parents table**
- parent_id: parent's ID, datatype INT and can't be null
- mother_name: mother's name, datatype VARCHAR(25) but can be null if diseased
- father:_name father's name, datatype VARCHAR(25) but can be null if diseased
- parent_phone: parent's phone number, datatype VARCHAR(10) and can't be null
- parent_address: parent's address, datatype VARCHAR(100)

4. **enrollments table**
- course_id: datatype INT and will be used in database's relationship
- stu_id: datatype INT and will be used in database's relationship
- enroll_id: enrollment's ID, datatype INT and can't be null
- enroll_date: enrollment's date, datatype DATE and can't be null

5. **courses table**
- course_id: course's ID, datatype INT and can't be null
- course_name: course's name, datatype VARCHAR(25) and can't be null
- course_dur: course's duration, datatype TINYINT and can't be null
- course_des: course's description, datatype VARCHAR(100) can be null

6. **instructors table**
- sub_id: datatype INT and will be used in database's relationship
- teach_id: datatype INT and will be used in database's relationship
- instruct_id: instructor's ID, data type INT and can't be null

7. **subjects table**

- sub_id: subject's ID, datatype INT and can't be null
- sub_name: subject's name, datatype VARCHAR(25) and can't be null
- sub_credit: subject's total credit that students need to take, datatype TINYINT and can't be null
- sub_startdate: first day of the subject, datatype DATE and can't be null
- sub_enddate: last day of the subject, datatype DATE and can't be null
- course_id: datatype INT and will be used in database's relationship

8. **grades table**

- grade_id: grade's ID, datatype INT and can't be null
- grade_score: the given score on a particular subject, datatype TINYINT and can't be null
- grade_year: the academic year of the grade, datatype TINYINT and can't be null
- grade_semester: grade's semester either 1 or 2, datatype TINYINT and can't be null
- sub_id: datatype INT and will be used in database's relationship
- stu_id: datatype INT and will be used in database's relationship

9. **teachers table**

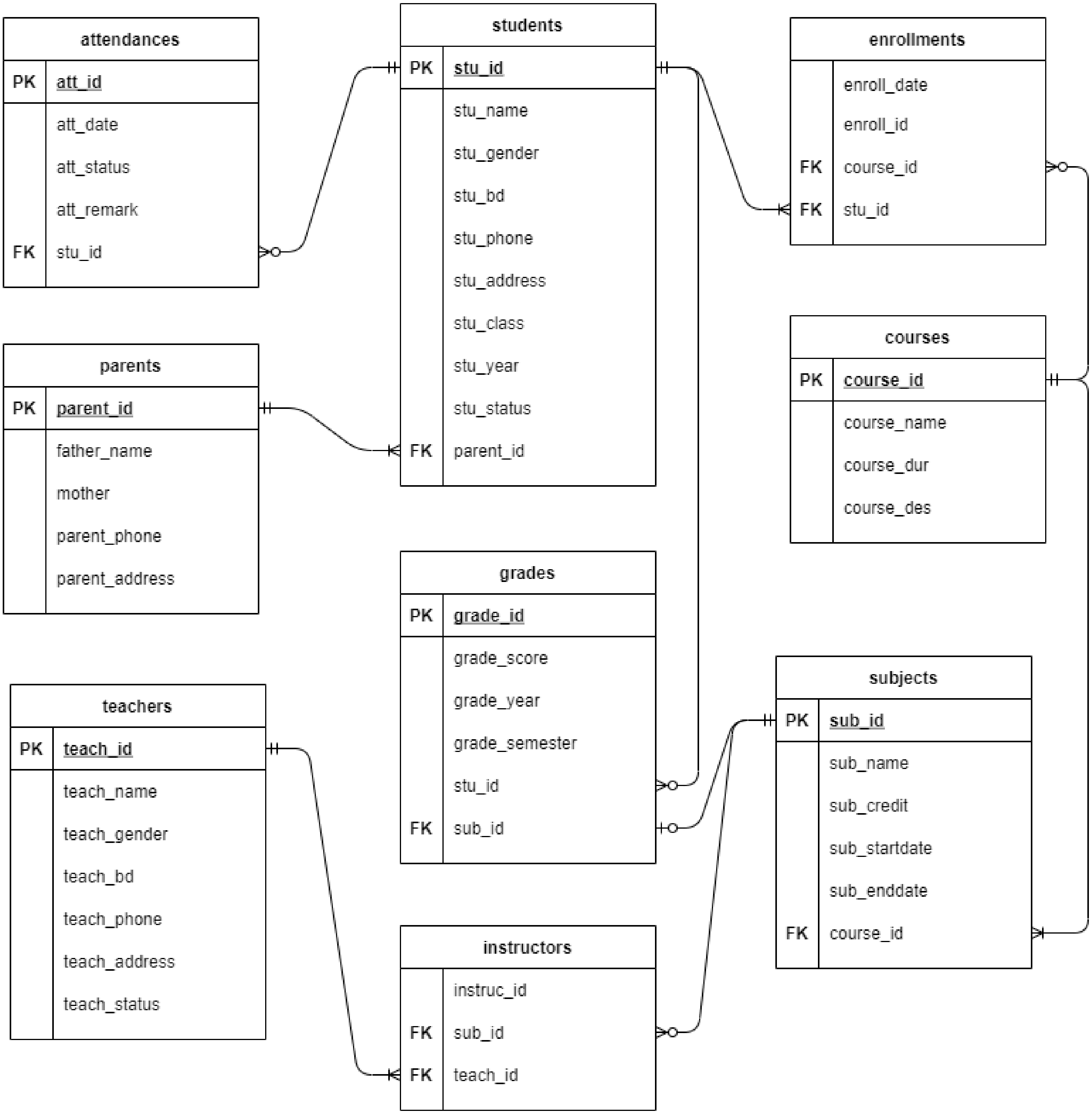
- teach_id: teacher's ID, datatype INT and can't be null
- teach_name: teacher's name, datatype VARCHAR(25) and can't be null
- teach_gender: teacher's gender, datatype CHAR(1) but can be null
- teach_bd: teacher's birthday, datatype DATE and can be null
- teach_phone: teacher's phone number, datatype VARCHAR(10) and can't be null
- teach_address: teacher's address, datatype VARCHAR(150) can be null
- teach_status: datatype BIT and should 1 if the teacher is still teaching, can't be null.

2.3 CONSTRAINTS

We've categorized the data into multiple tables to further improve our database's data organization and management, but there's still one more step to complete our design. Having a well-structured organization is nice, but giving those data the ability to reference each other is more preferable. That is when constraints come into play. Constraints are the core fundamentals in relational database design and without them, our well-organized data wouldn't be able to communicate, making them unmanageable.

We'll start by adding primary keys and foreign keys to our tables to enforce table relationship allowing all the tables to communicate with each other. This will in turn, give us the ability to perform advances operations in database such as joining tables for instances. That's why our team have made an **Entity Relationship Diagram** to demonstrate our database's table relationships.

School Database ERD



As shown in the diagram above:

- **parents** has parent_id as its primary key and has one to many relationship with students
- **attendances** has att_id as its primary key and stu_id as its foreign key
- **enrollments** has course_id and stu_id as its foreign keys with no primary key
- **students** has stu_id as the primary key and parent_id as the foreign key, it also has one to many relationship with **enrollments** and **grades**
- **courses** has the primary key of course_id, and it has one to many relationship with **enrollments** and one to many relationship with **subjects**
- **subjects** has sub_id as its primary key and also have one to one relationship with **grades** and one to many relationship with **instructors**
- **grades** has grade_id as the primary key, while sub_id and stu_id are foreign keys.
- **instructors** has sub_id and teach_id as its foreign keys with no primary key
- **teachers** has teach_id as the primary key.

Among these table relationships, it's also important to note that:

- **students** table has many to many relationship with **courses** table as a student can register for multiple courses and one course can also have many students, it was made possible due to the creation of **enrollments** table
- **teachers** and **subjects** is also a many to many relationship because a teacher can't teach multiple subjects and a subject can be also taught by multiple teachers, and **instructors** table is a link between the two tables.
- Additional constraints will also be added to reinforce referential integrity in our database to ensure data consistencies among all the tables. We'll be adding more constraints to prevent unauthorized insertion or deletion. On top of that, we'll also add clustered and non-clustered in indexes to improve our database speed when retrieving data. Keep all the data safe while maintaining speed is always our priority and we'll be doing our best to improve referential integrity and prevent any data violation with the constraints.

2.4 TESTING DATA

While database design is probably one of the most important part of system analysis, we will also need data before we get into system implementation. Finding data beforehand might pave the way for the following stage as we will get more time to focus only on building our system rather wasting time finding data to test the database. Moreover, it can save us more testing time giving our team more opportunity to improve our system. That's the reason why, we have gathered serval data to prepare for system implementation.

1. **courses**

courses_id	courses_name	courses_dur	courses_des
1	Information Technology	4	
2	Mathematic	4	
3	English Literature	4	
4	Khmer Literature	4	
5	Physics	4	
6	International Relation	4	
7	Tourism	4	
8	Business	4	
9	Management	4	
10	Chemistry	4	
11	Biology	4	
12	Environment Science	4	
13	Sociology	4	
14	Psychology	4	
15	Accounting	4	
16	Linguistics	4	
17	Information Engineering	4	
18	Philosophy	4	
19	Geography	4	
20	Photography	4	

2. **teachers**

teach_id	teach_name	teach_gender	teach_bd	teach_phone	teach_address	teach_st
1000001	Hun Sroy	M	1988-01-28	010251699	Phnom Penh	1
1000002	Kan Leang	M	1979-02-19	016125899	Kampong Cham	1
1000003	So Pheak	M	1995-03-13	092888799	Phnom Penh	1
1000004	Thea Srun	M	1978-04-18	08828899	Battambang	1
1000005	Vy Thida	F	1981-05-06	011231892	Banteay Meanchey	1
1000006	Dy Mangky	M	1988-06-09	012888859	Phnom Penh	1
1000007	Chan Dara	M	1985-02-12	012356865	Phnom Penh	1
1000008	Chhom Davit	M	1987-12-30	015522525	Phnom Penh	1
1000009	Thy Danuth	M	1991-08-12	012250654	Prey Veng	1
1000010	Kong Lida	F	1973-05-01	015580425	Phnom Penh	1
1000011	Long Rathana	M	1899-02-20	011522501	Phnom Penh	1
1000012	Lin Darith	M	1990-02-20	010020302	Phnom Penh	1
1000013	Kin Dina	F	1994-04-30	098474521	Phnom Penh	1
1000014	Lin Naroth	M	1994-12-06	098461521	Phnom Penh	1
1000015	Yin Vanuth	M	1989-04-04	012580656	Kampong Cham	1
1000016	Dy Rotha	F	1991-02-06	093543524	Phnom Penh	1
1000017	Yuth Chivon	M	1978-08-25	012823871	Takav	1
1000018	Hong Vitou	M	1987-12-30	016557854	Battambang	1
1000019	Chea Davin	M	1977-04-11	012542487	Phnom Penh	1
1000020	Chak Riya	F	1992-12-05	092518792	Phnom Penh	1

3. students

stu_id	stu_name	stu_ge	stu_bd	stu_pho	stu_address	stu_cl	stu_ye	stu_s	parent
1000001	Bo Phanit	M	1999-01-09	097123456	Phnom Penh	B107	2	1	1000006
1000002	Pal Bora	M	2000-02-08	070133456	Battambang	B107	2	1	1000010
1000003	Nget Sayha	M	2001-03-07	097143456	Phnom Penh	B107	2	1	1000005
1000004	Mut Tola	M	2000-04-06	077153456	Phnom Penh	B107	2	1	1000004
1000005	Srey Lin	F	2000-05-05	097163456	Takeo	B107	2	1	1000001
1000006	Ma Brosa	F	2000-06-04	092173456	Phnom Penh	B107	2	1	1000020
1000007	Reach Bot	M	2000-07-03	097183456	Battambang	B107	2	1	1000002
1000008	Kao Kanha	F	2000-08-27	012193456	Phnom Penh	B107	2	1	1000003
1000009	So Polin	M	2000-09-01	088103456	Phnom Penh	B107	2	1	1000015
1000010	So Visal	M	2000-10-11	097123416	Battambang	B107	2	1	1000011
1000011	Meas Kakda	M	1998-11-20	097123426	Phnom Penh	B107	2	1	1000014
1000012	Sin Sophea	F	2000-12-12	016123436	Phnom Penh	B107	2	1	1000013
1000013	Ly Ta	F	2001-01-01	097123446	Phnom Penh	B107	2	1	1000012
1000014	Seam Sothy	M	2001-02-02	093123466	Phnom Penh	B107	2	1	1000018
1000015	Thean Long	M	2001-03-03	097123476	Takeo	B107	2	1	1000009
1000016	Mao Sola	M	2001-04-14	010123486	Phnom Penh	B107	2	1	1000017
1000017	Vi Lady	F	2001-05-05	097123496	Phnom Penh	B107	2	1	1000007
1000018	Sen Mesa	M	2001-07-18	097123406	Battambang	B107	2	1	1000016
1000019	Ma Sudan	F	2001-04-14	096123457	Phnom Penh	B107	2	1	1000008
1000020	Nhom China	M	2001-05-24	097123458	Phnom Penh	B107	2	1	1000019

4. subjects

sub_id	sub_name	sub_credit	sub_startdate	sub_enddate	course_id
1	C Programming 1	25	2019-11-06	2020-03-05	1
2	Fundamental Computer 1	25	2019-11-09	2020-03-04	1
3	Mathematics 1	25	2019-11-07	2020-03-05	1
4	Electronic 1	25	2019-11-07	2020-03-07	1
5	21st Century	25	2019-11-06	2020-03-04	1
6	English for Computer 1	25	2019-11-08	2020-03-08	1
7	C Programming 2	25	2020-07-28	2020-11-03	1
8	Fundamental Computer 2	25	2020-07-27	2020-11-02	1
9	Electronic 2	25	2020-07-30	2020-11-05	1
10	Khmer civilization	25	2020-07-29	2020-11-04	1
11	History	25	2020-07-27	2020-11-02	1
12	Mathematic 2	25	2020-07-28	2020-11-03	1
13	English 2	25	2020-07-27	2020-11-06	1
14	Database System 1	25	2020-12-24	2021-04-22	1
15	C++ Programming 1	25	2020-12-24	2021-04-22	1
16	Computer Architecture 1	25	2020-12-21	2021-04-19	1
17	Data structure 1	25	2020-12-23	2021-04-21	1
18	Data communication 1	25	2020-12-21	2021-04-19	1
19	English 3	25	2020-12-22	2021-04-20	1
20	Statistic Math 1	25	2019-11-06	2020-03-05	2

5. grades

grade_id	grade_score	grade_year	grade_semester	sub_id	stu_id
1	90	2	1	14	1000001
2	73	2	1	14	1000002
3	92	2	1	14	1000003
4	86	2	1	14	1000004
5	95	2	1	14	1000005
6	81	2	1	14	1000006
7	82	2	1	14	1000007
8	74	2	1	14	1000008
9	81	2	1	14	1000009
10	96	2	1	14	1000010
11	88	2	1	14	1000011
12	90	2	1	14	1000012
13	87	2	1	14	1000013
14	70	2	1	14	1000014
15	90	2	1	14	1000015
16	87	2	1	14	1000016
17	92	2	1	14	1000017
18	73	2	1	14	1000018
19	85	2	1	14	1000019
20	90	2	1	14	1000020

6. parents

parent_id	father	mother	parent_phone	Par_address
1000001	Chan Sopheak	So Lina	0979869751	Phnom Penh
1000002	Sorn Dara	Heng Dalin	098964512	Kampot
1000003	Toch Vannut	Sok Sreynit	0978732324	Kampong Cham
1000004	Ly Mengleang	Heng Sevmei	098786898	Phnom Penh
1000005	Eng Davith	Soth Kaknika	098246679	Phnom Penh
1000006	Som Narith	Van Dina	086836497	Phnom Penh
1000007	Vuth Oudom	Chan Sreyneang	0976828690	Phnom Penh
1000008	Vann Din	Chhang Lyly	0972342423	Kampot
1000009	Chhun Davit	Heng Reaksmey	087123741	Kampong Thom
1000010	Hort Ratana	Ear Hengly	083479713	Kampong Cham
1000011	Lun Kimheng	Lay Dary	012372342	Phnom Penh
1000012	Kan Chetra	Sim Sina	092414144	Phnom Penh
1000013	Choun Visal	Srang Lika	098213445	Phnom Penh
1000014	Seng Vibol	Heng Menghorn	099761334	Kampong Cham
1000015	Try Kimseng	Ly Yanet	012243432	Phnom Penh
1000016	Hun Davin	Si Sreytoch	098348934	Phnom Penh
1000017	Van Henglong	Sreang Sreylin	095634223	Phnom Penh
1000018	An Vandet	Leng Sinat	076435623	Phnom Penh
1000019	Tha Darith	Ly Kunthea	0971343134	Phnom Penh
1000020	Sorn Vannak	Pich Monita	099112345	Phnom Penh

7. **enrollments**

enroll_id	Enroll_date	course_id	stu_id
1	2019-09-25	1	1000001
2	2019-09-26	1	1000002
3	2019-09-25	1	1000003
4	2019-09-26	1	1000004
5	2019-09-28	1	1000005
6	2019-09-28	1	1000006
7	2019-09-28	1	1000007
8	2019-10-01	1	1000008
9	2019-10-03	1	1000009
10	2019-10-06	1	1000010
11	2019-10-06	1	1000011
12	2019-10-07	1	1000012
13	2019-10-10	1	1000013
14	2019-10-12	1	1000014
15	2019-10-13	1	1000015
16	2019-10-13	1	1000016
17	2019-10-17	1	1000017
18	2019-10-18	1	1000018
19	2019-10-25	1	1000019
20	2019-11-01	1	1000020

8. **instructors**

instruc_id	teach_id	sub_id
1	1000006	14
2	1000007	14
3	1000008	14
4	1000003	15
5	1000002	15
6	1000004	16
7	1000005	16
8	1000001	15
9	1000017	17
10	1000018	18
11	1000019	18
12	1000015	17
13	1000014	19
14	1000013	19
15	1000016	17
16	1000009	16
17	1000020	18
18	1000012	19
19	1000010	1
20	1000011	1

9. attendances

att_id	att_date	att_status	att_remark	stu_id
1	2020-12-24 17:00:00	1		1000001
2	2020-12-24 17:00:00	1		1000002
3	2020-12-24 17:00:00	1		1000003
4	2020-12-24 17:00:00	1		1000004
5	2020-12-24 17:00:00	0	Permission	1000005
6	2020-12-24 17:00:00	1		1000006
7	2020-12-24 17:00:00	0		1000007
8	2020-12-24 17:00:00	0		1000008
9	2020-12-24 17:00:00	1		1000009
10	2020-12-24 17:00:00	1		1000010
11	2020-12-24 17:00:00	1		1000011
12	2020-12-24 17:00:00	1		1000012
13	2020-12-24 17:00:00	1		1000013
14	2020-12-24 17:00:00	0	Permission	1000014
15	2020-12-24 17:00:00	1		1000015
16	2020-12-24 17:00:00	1		1000016
17	2020-12-24 17:00:00	1		1000017
18	2020-12-24 17:00:00	1		1000018
19	2020-12-24 17:00:00	1	Late	1000019
20	2020-12-24 17:00:00	1	Late	1000020

2.5 SUMMARY

In conclusion, we've looked at tables that are needed for the database and we also saw the attributes within each table. We've covered several constraints specifically tables' relationships and more constraints will be added later in system implementation. As previously mentioned, we also need to add some fake data to pave the way for the next stage and to save us time. These sum up our plan and system analysis for our database. We're looking forward to implement these into our final product.

III. SYSTEM IMPLEMENENATION

So far, we've only looked at all the important aspects needed to build our database. With our system analysis, we were able to virtualize and fully understand what our database look like and how it should be. Our team's next mission is to make our system analysis a reality. We'll starting by creating tables for our database. Then we'll include all the essential constraints to create a relational database. Lastly, our team will insert our tested data into the database in order to examine our system.

3.1 CREATING TABLES

We'll begin by creating a database called school_management_system using a simple create query:

```
CREATE DATABASE school_management_system;
```

After our database has been created. We're ready to add more tables into it. As a reminder, we had 9 tables that should be included inside our database. We'll start by creating them one at time using separated query based on our system analysis.

1. **Creating parents table:**

```
CREATE TABLE parents(
       parent_id INT NOT NULL IDENTITY(1000001, 1),
       father_name VARCHAR(25),
       mother_name VARCHAR(25),
       parent_phone VARCHAR(10),
       parent_address VARCHAR(100)

);
```

2. **Creating teachers table:**

```
CREATE TABLE teachers(
       teach_id INT NOT NULL IDENTITY(1000001, 1),
       teach_name VARCHAR(25) NOT NULL,
       teach_gender CHAR(1),
       teach_bd DATE NOT NULL,
       teach_phone VARCHAR(10) NOT NULL,
       teach_address VARCHAR(100),
       teach_status BIT NOT NULL
);
```

3. **Creating subjects table:**

```
CREATE TABLE subjects(
       sub_id INT NOT NULL IDENTITY(1, 1),
       sub_name VARCHAR(25) NOT NULL,
       sub_credit TINYINT NOT NULL,
       sub_startdate DATE NOT NULL,
       sub_enddate DATE NOT NULL,
       course_id INT NOT NULL
);
```

4. Creating students table:

```
CREATE TABLE students(
        stu_id INT NOT NULL IDENTITY(1000001, 1),
        stu_name VARCHAR(25) NOT NULL,
        stu_gender CHAR NULL,
        stu_bd DATE NOT NULL,
        stu_phone VARCHAR(10),
        stu_address VARCHAR(100),
        stu_class CHAR(4) NOT NULL,
        stu_year SMALLINT NOT NULL,
        stu_status BIT NOT NULL,
        parent_id INT NOT NULL

);
```

5. Creating courses table:

```
CREATE TABLE courses(
        course_id INT NOT NULL IDENTITY(1, 1),
        course_name VARCHAR(25) NOT NULL,
        course_dur TINYINT NOT NULL,
        course_des VARCHAR(100)
);
```

6. Creating enrollments table:

```
CREATE TABLE enrollments(
        enroll_id INT NOT NULL IDENTITY(1, 1),
        enroll_date DATETIME NOT NULL,
        stu_id INT NOT NULL,
        course_id INT NOT NULL
);
```

7. Creating attendances table:

```
CREATE TABLE attendances(
        att_id INT NOT NULL IDENTITY(1, 1),
        att_date DATETIME NOT NULL,
        att_status BIT NOT NULL,
        att_remark VARCHAR(25),
        stu_id INT NOT NULL
);
```

8. Creating **grades** table:

```
CREATE TABLE grades(
        grade_id INT NOT NULL IDENTITY(1, 1),
        grade_score TINYINT NOT NULL,
        grade_year SMALLINT NOT NULL,
        grade_semester TINYINT NOT NULL,
        sub_id INT NOT NULL,
        stu_id INT NOT NULL
);
```

9. Creating instructors table:

```
CREATE TABLE instructors(
        instruc_id INT NOT NULL IDENTITY(1, 1),
        sub_id INT NOT NULL,
        teach_id INT NOT NULL
);
```

3.2 ADDING CONSTRAINTS

Up to this point, we've just created the necessary tables for our database, but we haven't added any constraints to any column in the table. We believe it's better to add constraints in a separate part for better organization and readability. Our goal is to deliver a clear and effective message in order to avoid and save the readers from all kind of confusions and misunderstandings. That's the reason why we decided to add constraints for each table separately in `ALTER TABLE` instead of adding them at the time the tables were created, and we hope this will help readers understand as much as we want it to be.

3.2.1 Defining primary keys

We'll be adding index to our primary key constraints in order to improve our database speed when searching or retrieving data and to save up space, we'll add them at the same in a single `ALTER TABLE`.

- Adding primary key and index for **students** table:

```
ALTER TABLE students WITH NOCHECK
ADD CONSTRAINT PK_students PRIMARY KEY CLUSTERED (stu_id);
```

- Adding primary key and index for **attendances** table:

```
ALTER TABLE attendances WITH NOCHECK
ADD CONSTRAINT PK_attendances PRIMARY KEY CLUSTERED (att_id);
```

- Adding primary key and index for **courses** table:

```
ALTER TABLE courses WITH NOCHECK
ADD CONSTRAINT PK_courses PRIMARY KEY CLUSTERED (course_id);
```

- Adding primary key and index for **enrollments** table:

```
ALTER TABLE enrollments WITH NOCHECK
ADD CONSTRAINT PK_enrollments PRIMARY KEY CLUSTERED (enroll_id);
```

- Adding primary key and index for **grades** table:

```
ALTER TABLE grades WITH NOCHECK
ADD CONSTRAINT PK_grades PRIMARY KEY CLUSTERED (grade_id);
```

- Adding primary key and index for **instructors** table:

```
ALTER TABLE instructors WITH NOCHECK
ADD CONSTRAINT PK_instructors PRIMARY KEY CLUSTERED (instruc_id);
```

- Adding primary key and index for **parents** table:

```
ALTER TABLE parents WITH NOCHECK
ADD CONSTRAINT PK_parents PRIMARY KEY CLUSTERED (parent_id);
```

- Adding primary key and index for **subjects** table:

```
ALTER TABLE subjects WITH NOCHECK
ADD CONSTRAINT PK_subjects PRIMARY KEY CLUSTERED (sub_id);
```

- Adding primary key and index for **teachers** table:

```
ALTER TABLE teachers WITH NOCHECK
ADD CONSTRAINT PK_teachers PRIMARY KEY CLUSTERED (teach_id);
```

3.2.2 Defining foreign keys

Just like we did when we added primary keys to our tables, we'll be adding non-clustered index in a single query.

- Adding foreign key and index to **students** table:

```
ALTER TABLE students
ADD CONSTRAINT FK_students_parents FOREIGN KEY (parent_id)
       REFERENCES parents (parent_id) ON DELETE NO ACTION;
CREATE NONCLUSTERED INDEX IX_students_parent_id
ON students (parent_id);
```

- Adding foreign key and index to **attendances** table:

```
ALTER TABLE attendances
ADD CONSTRAINT FK_attendances_students FOREIGN KEY (stu_id)
       REFERENCES students (stu_id) ON DELETE NO ACTION;
CREATE NONCLUSTERED INDEX IX_attendances_stu_id
ON attendances (stu_id);
```

- Adding foreign key and index to **enrollments** table:

```
ALTER TABLE enrollments
ADD CONSTRAINT FK_enrollments_students FOREIGN KEY (stu_id)
              REFERENCES students (stu_id) ON DELETE NO ACTION,
       CONSTRAINT FK_enrollments_courses FOREIGN KEY (course_id)
              REFERENCES courses (course_id) ON DELETE NO ACTION;
CREATE NONCLUSTERED INDEX IX_enrollments_sub_id_course_id
ON enrollments (stu_id, course_id);
```

- Adding foreign key and index to **subjects** table:

```
ALTER TABLE subjects
ADD CONSTRAINT FK_subjects_courses FOREIGN KEY (course_id)
       REFERENCES courses (course_id) ON DELETE NO ACTION;
CREATE NONCLUSTERED INDEX IX_subjects_course_id
ON subjects (course_id);
```

- Adding foreign key and index to **grades** table:

```
ALTER TABLE grades
ADD CONSTRAINT FK_grades_subjects FOREIGN KEY (sub_id)
              REFERENCES subjects (sub_id) ON DELETE NO ACTION,
       CONSTRAINT FK_grades_students FOREIGN KEY (stu_id)
              REFERENCES students (stu_id) ON DELETE NO ACTION;
CREATE NONCLUSTERED INDEX IX_grade_sub_id
ON grades (sub_id);
```

- Adding foreign key and index to **instructors** table:

```
ALTER TABLE instructors
ADD CONSTRAINT FK_instructors_subjects FOREIGN KEY (sub_id)
              REFERENCES subjects (sub_id) ON DELETE NO ACTION,
       CONSTRAINT FK_instructors_teachers FOREIGN KEY (teach_id)
              REFERENCES teachers (teach_id) ON DELETE NO ACTION;
CREATE NONCLUSTERED INDEX IX_instructors_sub_id_teach_id
ON instructors (sub_id, teach_id);
```

3.3 INSERTING DATA

In system analysis, we created some data for testing. For this part, we'll be adding them in our tables with each table containing 20 rows of data. Because adding all the insert queries in one place takes a lot of space and looks complicated. To improve readability, our team think it's best for them to stay in separate pages. We'll add those queries to the appendix and includes the hyperlinks to each page to make it easy for reader to follow through and at the same time, make things look cleaned and organized.

1. Inserting data into **courses** table
2. Inserting data into **parents** table
3. Inserting data into **students** table
4. Inserting data into **attendances** table
5. Inserting data into **enrollments** table
6. Inserting data into **subjects** table
7. Inserting data into **grades** table
8. Inserting data into **teachers** table
9. Inserting data into **instructors** table

3.4 SUMMARY

In short, we've created our database called school_management_system. We also added primary key and foreign key constraints as well as inserting some data into each table. We've built everything from scratch and now what's left is just testing. Our team have put all of our best efforts to optimize our database as much as possible.

However, implementing our database isn't the end of the journey, many features still need to be added in the future to extend its capabilities. There're always rooms for improvements, and there're always rooms for optimizations in our database.

IV. REPORT

4.1 PROJECT DESCRIPTION

CLASS	GROUP	ACADEMIC YEAR
Class E4	Group 1	02nd Year, 23rd Generation
PROJECT NAME	PROJECT DURATION	PREPARED BY
School Management System	5 weeks	Seng Chetra

Our school database built specifically for students with the purpose of

1. Improving their learning experience as well as providing a modern learning approach, which give freedom and allow them to view their grades with ease at anytime, anywhere
2. Providing the school, a better learning quality to rival other competitors

4.2 PROJECT OVERVIEW

TASK	% DONE	DUE DATE	NOTES
Project proposal	100	March 06th, 2021	
System analysis	100	March 20th, 2021	
System implementation	90	April 03rd, 2021	More data was needed
API implementation	80	April 14th, 2021	Basic API, more tests are required
Interface implementation	2	April 14th, 2021	Deadline was shifted and more time was needed
Project Report	100	April 25th, 2021	Deadline was shifted

4.3 INDIVIDUAL OVERVIEW

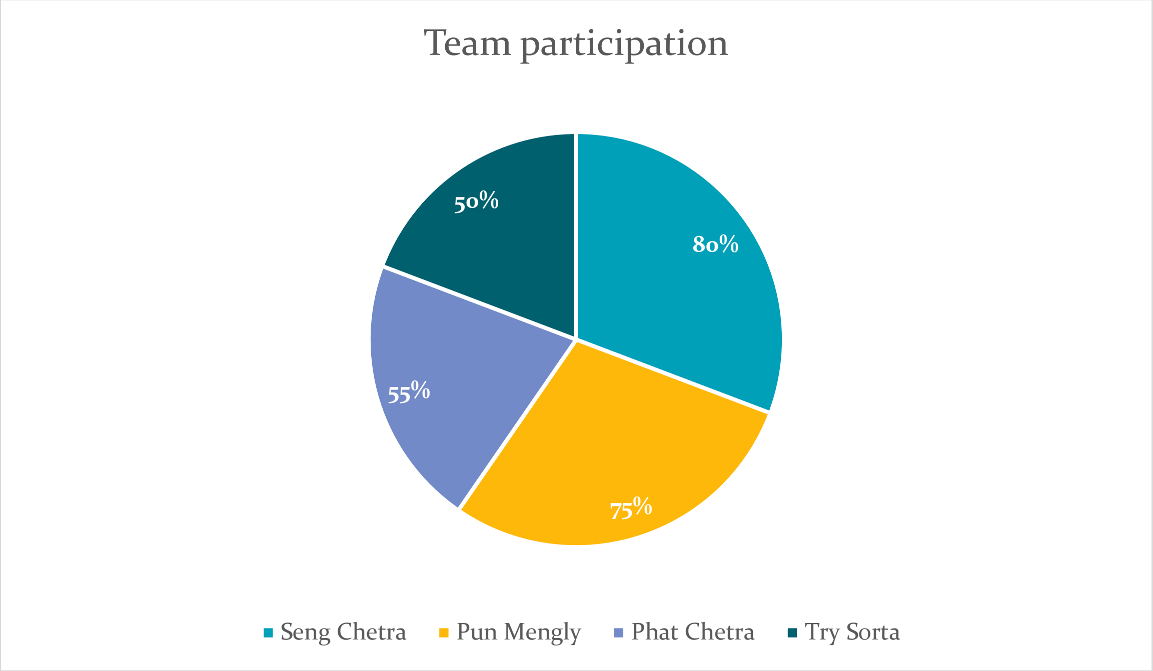
NAME	ROLE
Seng Chetra	Project Manager and Database designer
Pun Mengly	Database designer and API implementer
Phat Chetra	Data implementer and Tester
Try Sorta	Data implementer and Tester

Documenting and project planning project manager's duties, which means proposal, system analysis, report and other important documents were all done and written by the project manager. It's essential to avoid working under pressure. Therefore, advanced planning was required and it was made ahead of time to ensure almost if not all the required tasks are completed before the deadline. System analysis was made as rules, so all group members can follow and completed their work accordingly. All documents from each stage including the final documents were made as a representation and as a guide to the audiences and readers to show them what and how our project was made. Moreover, Project schedule and project road map was set by project manager in order to improve work productivities and be able to succeed with the given short amount of time. These task below were given to each group member according to their roles:

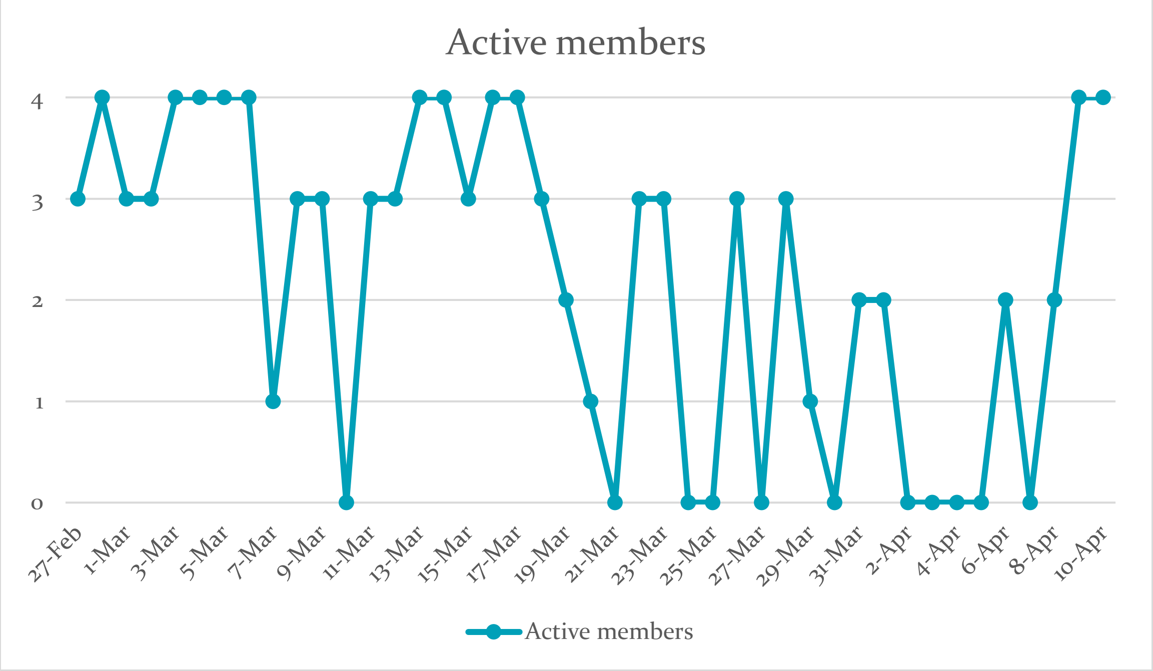
TASK	ASSIGNED DATE	ASSIGNED TO
Project planning	February 27th, 2021	Seng Chetra
Proposal document	March 01st, 2021	Seng Chetra
Proposal document review	March 06th, 2021	Pun Mengly, Phat Chetra, Try Sorta
System analysis document	March 08th, 2021	Seng Chetra
API implementation	March 10th, 2021	Pun Mengly
Database design	March 10th, 2021	Seng Chetra
Database and ERD review	March 14th, 2021	Pun Mengly
Generating testing data	March 17th, 2021	Phat Chetra, Try Sorta
System analysis document review	March 20th, 2021	Pun Mengly, Phat Chetra, Try Sorta
System implementation	March 27th, 2021	Seng Chetra
System implementation document	March 27th, 2021	Seng Chetra
System implementation document review	March 27th, 2021	Phat Chetra, Pun Mengly
Database modification	April 06th, 2021	Pun Mengly
System testing	April 08th, 2021	Phat Chetra, Try Sorta
Project report	April 08th, 2021	Seng Chetra
Project report review	April 09th, 2021	Pun Mengly

2.5 GROUP ACTIVITIES SUMMARY

Our team members were active and had great impacts on the overall project. Everyone has taken part in it and they completed their assigned tasks with no or less issues. The chart below shows our team members participation:



The graph below shows the number of active members each day since the day our team was formed till the final day of the project:



2.6 PROJECT ISSUES

Although we were able to complete our project with the given time, there are still some issues that are needed to addressed. Memory and database optimizations for example, are the future issues we haven't discuss. Our database has been optimized for now to handle just enough data, but that doesn't mean further adjustments are not required. If we take a look at the problems that we'd faced along the way when building our project, there were a few that caused delays and difficulties.

ISSUE	REASON
Time	Due to the examination date changes made by the school, we had less time work to with our project
Experience	This our first time making our database project, so there were a lot of things that we didn't know could be done better
Research	With limited time, we haven't able to do more research and add advance features into our database
Quarantine	Online meeting didn't deliver the best experience and it was the only option available to us

2.7 FUTURE IMPACTS

During the process of making this project possible, we've gained a lot of bittersweet experiences, some were fun and great, others were stressful and unpleasant. However, it was a great opportunity being able to work with people and learn a lot from them. Furthermore, we look forward to working with this project again in the near future because if given enough time and resources, it could provide a lot of benefits to students such as

- Revolutionizing our learning experience
- Providing convenience to many students from middle school all the way up to university students
- Motivating students to use technologies and be familiar with it
- Assisting them with studies, and further improve their understanding of computers, databases, securities and so many more.

With these wonderful advantages, more time and resources should be invested into this project and more efforts have to be made to achieve the best possible result, and get the most out of it. We're certain that this project has the potential to improve the education standard as a whole once it's been fully finalized.

2.8 FEEDBACKS

We've come a long way, and all great things must come to an end. We personally think this project was a bit challenging because of the busy schedules that everyone has to deal with, and the pressure of not being able to have a face to face meeting during this quarantine. However, at the end of day, we did gain a lot from it, we learned so many things, get to know more people and work together online to achieve the same goal.

Our final thoughts about the project are positive. Its aims were stated clearly and every step and requirement to achieve it was detailed, so we didn't have issues trying understand what we had to do in each stage. We also think it was advantageous to us because the project taught us a lot more than a normal project would have about building our own databases, and we got to experience what it feels like to work as a real developer. We're also excited and we can't wait to work on more projects like this in the future.

Appendix

```sql
INSERT INTO courses(course_name, course_dur, course_des)
VALUES ('Information Technology', 4, NULL);

INSERT INTO courses(course_name, course_dur, course_des)
VALUES ('Mathematic', 4, NULL);

INSERT INTO courses(course_name, course_dur, course_des)
VALUES ('English Literature', 4, NULL);

INSERT INTO courses(course_name, course_dur, course_des)
VALUES ('Khmer Literature', 4, NULL);

INSERT INTO courses(course_name, course_dur, course_des)
VALUES ('Physics', 4, NULL);

INSERT INTO courses(course_name, course_dur, course_des)
VALUES ('International Relation', 4, NULL);

INSERT INTO courses(course_name, course_dur, course_des)
VALUES ('Tourism', 4, NULL);

INSERT INTO courses(course_name, course_dur, course_des)
VALUES ('Business', 4, NULL);

INSERT INTO courses(course_name, course_dur, course_des)
VALUES ('Management', 4, NULL);

INSERT INTO courses(course_name, course_dur, course_des)
VALUES ('Chemistry', 4, NULL);

INSERT INTO courses(course_name, course_dur, course_des)
VALUES ('Biology', 4, NULL);

INSERT INTO courses(course_name, course_dur, course_des)
VALUES ('Environment Science', 4, NULL);

INSERT INTO courses(course_name, course_dur, course_des)
VALUES ('Sociology', 4, NULL);

INSERT INTO courses(course_name, course_dur, course_des)
VALUES ('Psychology', 4, NULL);

INSERT INTO courses(course_name, course_dur, course_des)
VALUES ('Accounting', 4, NULL);

INSERT INTO courses(course_name, course_dur, course_des)
VALUES ('Linguistics', 4, NULL);

INSERT INTO courses(course_name, course_dur, course_des)
VALUES ('Information Engineering', 4, NULL);

INSERT INTO courses(course_name, course_dur, course_des)
VALUES ('Philosophy', 4, NULL);

INSERT INTO courses(course_name, course_dur, course_des)
VALUES ('Geography', 4, NULL);
```

```
INSERT INTO courses(course_name, course_dur, course_des)
VALUES ('Photography', 4, NULL);
```

```
INSERT INTO parents(father_name, mother_name, parent_phone, parent_address)
VALUES ('Chan Sopheak', 'So Lina', '0979869751', 'Phnom Penh');

INSERT INTO parents(father_name, mother_name, parent_phone, parent_address)
VALUES ('Sorn Dara', 'Heng Dalin', '098964512', 'Kampot');

INSERT INTO parents(father_name, mother_name, parent_phone, parent_address)
VALUES ('Toch Vannut', 'Sok Sreynit', '0978732324', 'Kampong Cham');

INSERT INTO parents(father_name, mother_name, parent_phone, parent_address)
VALUES ('Ly Mengleang', 'Heng Sevmei', '098786898', 'Phnom Penh');

INSERT INTO parents(father_name, mother_name, parent_phone, parent_address)
VALUES ('Eng Davith', 'Soth Kaknika', '098246679', 'Phnom Penh');

INSERT INTO parents(father_name, mother_name, parent_phone, parent_address)
VALUES ('Som Narith', 'Van Dina', '086836497', 'Phnom Penh');

INSERT INTO parents(father_name, mother_name, parent_phone, parent_address)
VALUES ('Vuth Oudom', 'Chan Sreyneang', '0976828690', 'Phnom Penh');

INSERT INTO parents(father_name, mother_name, parent_phone, parent_address)
VALUES ('Vann Din', 'Chhang Lyly', '0972342423', 'Kampot');

INSERT INTO parents(father_name, mother_name, parent_phone, parent_address)
VALUES ('Chhun Davit', 'Heng Reaksmey', '087123741', 'Kampong Thom');

INSERT INTO parents(father_name, mother_name, parent_phone, parent_address)
VALUES ('Hort Ratana', 'Ear Hengly', '083479713', 'Kampong Cham');

INSERT INTO parents(father_name, mother_name, parent_phone, parent_address)
VALUES ('Lun Kimheng', 'Lay Dary', '012372342', 'Phnom Penh');

INSERT INTO parents(father_name, mother_name, parent_phone, parent_address)
VALUES ('Kan Chetra', 'Sim Sina', '092414144', 'Phnom Penh');

INSERT INTO parents(father_name, mother_name, parent_phone, parent_address)
VALUES ('Choun Visal', 'Srang Lika', '098213445', 'Phnom Penh');

INSERT INTO parents(father_name, mother_name, parent_phone, parent_address)
VALUES ('Seng Vibol', 'Heng Menghorn', '099761334', 'Kampong Cham');

INSERT INTO parents(father_name, mother_name, parent_phone, parent_address)
VALUES ('Try Kimseng', 'Ly Yanet', '012243432', 'Phnom Penh');

INSERT INTO parents(father_name, mother_name, parent_phone, parent_address)
VALUES ('Hun Davin', 'Si Sreytoch', '098348934', 'Phnom Penh');

INSERT INTO parents(father_name, mother_name, parent_phone, parent_address)
VALUES ('Van Henglong', 'Sreang Sreylin', '095634223', 'Phnom Penh');

INSERT INTO parents(father_name, mother_name, parent_phone, parent_address)
VALUES ('An Vandet', 'Leng Sinat', '076435623', 'Phnom Penh');

INSERT INTO parents(father_name, mother_name, parent_phone, parent_address)
VALUES ('Tha Darith', 'Ly Kunthea', '0971343134', 'Phnom Penh');
```

```sql
INSERT INTO parents(father_name, mother_name, parent_phone, parent_address)
VALUES ('Sorn Vannak', 'Pich Monita', '099112345', 'Phnom Penh');
```

```
INSERT INTO students(stu_name, stu_gender, stu_bd, stu_phone, stu_address, stu_class,
stu_year, stu_status, parent_id)
VALUES ('Bo Phanit', 'M', '1999-01-09', '097123456', 'Phnom Penh', 'B107', 2, 1,
1000006);

INSERT INTO students(stu_name, stu_gender, stu_bd, stu_phone, stu_address, stu_class,
stu_year, stu_status, parent_id)
VALUES ('Pal Bora', 'M', '2000-02-08', '070133456', 'Battambang', 'B107', 2, 1, 1000010);

INSERT INTO students(stu_name, stu_gender, stu_bd, stu_phone, stu_address, stu_class,
stu_year, stu_status, parent_id)
VALUES ('Nget Sayha', 'M', '2001-03-07', '097143456', 'Phnom Penh', 'B107', 2, 1,
1000005);

INSERT INTO students(stu_name, stu_gender, stu_bd, stu_phone, stu_address, stu_class,
stu_year, stu_status, parent_id)
VALUES ('Mut Tola', 'M', '2000-04-06', '077153456', 'Phnom Penh', 'B107', 2, 1, 1000004);

INSERT INTO students(stu_name, stu_gender, stu_bd, stu_phone, stu_address, stu_class,
stu_year, stu_status, parent_id)
VALUES ('Srey Lin', 'F', '2000-05-05', '097163456', 'Takeo', 'B107', 2, 1, 1000001);

INSERT INTO students(stu_name, stu_gender, stu_bd, stu_phone, stu_address, stu_class,
stu_year, stu_status, parent_id)
VALUES ('Ma Brosa', 'F', '2000-06-04', '092173456', 'Phnom Penh', 'B107', 2, 1, 1000020);

INSERT INTO students(stu_name, stu_gender, stu_bd, stu_phone, stu_address, stu_class,
stu_year, stu_status, parent_id)
VALUES ('Reach Bot', 'M', '2000-07-03', '097183456', 'Battambang', 'B107', 2, 1,
1000002);

INSERT INTO students(stu_name, stu_gender, stu_bd, stu_phone, stu_address, stu_class,
stu_year, stu_status, parent_id)
VALUES ('Kao Kanha', 'F', '2000-08-27', '012193456', 'Phnom Penh', 'B107', 2, 1,
1000003);

INSERT INTO students(stu_name, stu_gender, stu_bd, stu_phone, stu_address, stu_class,
stu_year, stu_status, parent_id)
VALUES ('So Polin', 'M', '2000-09-01', '088103456', 'Phnom Penh', 'B107', 2, 1, 1000015);

INSERT INTO students(stu_name, stu_gender, stu_bd, stu_phone, stu_address, stu_class,
stu_year, stu_status, parent_id)
VALUES ('So Visal', 'M', '2000-10-11', '097123416', 'Battambang', 'B107', 2, 1, 1000011);

INSERT INTO students(stu_name, stu_gender, stu_bd, stu_phone, stu_address, stu_class,
stu_year, stu_status, parent_id)
VALUES ('Meas Kakda', 'M', '1998-11-20', '097123426', 'Phnom Penh', 'B107', 2, 1,
1000014);

INSERT INTO students(stu_name, stu_gender, stu_bd, stu_phone, stu_address, stu_class,
stu_year, stu_status, parent_id)
VALUES ('Sin Sophea', 'F', '2000-12-12', '016123436', 'Phnom Penh', 'B107', 2, 1,
1000013);

INSERT INTO students(stu_name, stu_gender, stu_bd, stu_phone, stu_address, stu_class,
stu_year, stu_status, parent_id)
VALUES ('Ly Ta', 'F', '2001-01-01', '097123446', 'Phnom Penh', 'B107', 2, 1, 1000012);
```

```
INSERT INTO students(stu_name, stu_gender, stu_bd, stu_phone, stu_address, stu_class,
stu_year, stu_status, parent_id)
VALUES ('Seam Sothy', 'M', '2001-02-02', '093123466', 'Phnom Penh', 'B107', 2, 1,
1000018);

INSERT INTO students(stu_name, stu_gender, stu_bd, stu_phone, stu_address, stu_class,
stu_year, stu_status, parent_id)
VALUES ('Thean Long', 'M', '2001-03-03', '097123476', 'Takeo', 'B107', 2, 1, 1000009);

INSERT INTO students(stu_name, stu_gender, stu_bd, stu_phone, stu_address, stu_class,
stu_year, stu_status, parent_id)
VALUES ('Mao Sola', 'M', '2001-04-14', '010123486', 'Phnom Penh', 'B107', 2, 1, 1000017);

INSERT INTO students(stu_name, stu_gender, stu_bd, stu_phone, stu_address, stu_class,
stu_year, stu_status, parent_id)
VALUES ('Vi Lady', 'F', '2001-05-05', '097123496', 'Phnom Penh', 'B107', 2, 1, 1000007);

INSERT INTO students(stu_name, stu_gender, stu_bd, stu_phone, stu_address, stu_class,
stu_year, stu_status, parent_id)
VALUES ('Sen Mesa', 'M', '2001-07-18', '097123406', 'Battambang', 'B107', 2, 1, 1000016);

INSERT INTO students(stu_name, stu_gender, stu_bd, stu_phone, stu_address, stu_class,
stu_year, stu_status, parent_id)
VALUES ('Ma Sudan', 'F', '2001-04-14', '096123457', 'Phnom Penh', 'B107', 2, 1, 1000008);

INSERT INTO students(stu_name, stu_gender, stu_bd, stu_phone, stu_address, stu_class,
stu_year, stu_status, parent_id)
VALUES ('Nhom China', 'M', '2001-05-24', '097123458', 'Phnom Penh', 'B107', 2, 1,
1000019);
```

```sql
INSERT INTO attendances(att_date, att_status, att_remark, stu_id)
VALUES ('2020-12-24 17:00:00', 1, NULL, 1000001);

INSERT INTO attendances(att_date, att_status, att_remark, stu_id)
VALUES ('2020-12-24 17:00:00', 1, NULL, 1000002);

INSERT INTO attendances(att_date, att_status, att_remark, stu_id)
VALUES ('2020-12-24 17:00:00', 1, NULL, 1000003);

INSERT INTO attendances(att_date, att_status, att_remark, stu_id)
VALUES ('2020-12-24 17:00:00', 1, NULL, 1000004);

INSERT INTO attendances(att_date, att_status, att_remark, stu_id)
VALUES ('2020-12-24 17:00:00', 0, 'Permission', 1000005);

INSERT INTO attendances(att_date, att_status, att_remark, stu_id)
VALUES ('2020-12-24 17:00:00', 1, NULL, 1000006);

INSERT INTO attendances(att_date, att_status, att_remark, stu_id)
VALUES ('2020-12-24 17:00:00', 0, NULL, 1000007);

INSERT INTO attendances(att_date, att_status, att_remark, stu_id)
VALUES ('2020-12-24 17:00:00', 0, NULL, 1000008);

INSERT INTO attendances(att_date, att_status, att_remark, stu_id)
VALUES ('2020-12-24 17:00:00', 1, NULL, 1000009);

INSERT INTO attendances(att_date, att_status, att_remark, stu_id)
VALUES ('2020-12-24 17:00:00', 1, NULL, 1000010);

INSERT INTO attendances(att_date, att_status, att_remark, stu_id)
VALUES ('2020-12-24 17:00:00', 1, NULL, 1000011);

INSERT INTO attendances(att_date, att_status, att_remark, stu_id)
VALUES ('2020-12-24 17:00:00', 1, NULL, 1000012);

INSERT INTO attendances(att_date, att_status, att_remark, stu_id)
VALUES ('2020-12-24 17:00:00', 1, NULL, 1000013);

INSERT INTO attendances(att_date, att_status, att_remark, stu_id)
VALUES ('2020-12-24 17:00:00', 0, 'Permission', 1000014);

INSERT INTO attendances(att_date, att_status, att_remark, stu_id)
VALUES ('2020-12-24 17:00:00', 1, NULL, 1000015);

INSERT INTO attendances(att_date, att_status, att_remark, stu_id)
VALUES ('2020-12-24 17:00:00', 1, NULL, 1000016);

INSERT INTO attendances(att_date, att_status, att_remark, stu_id)
VALUES ('2020-12-24 17:00:00', 1, NULL, 1000017);

INSERT INTO attendances(att_date, att_status, att_remark, stu_id)
VALUES ('2020-12-24 17:00:00', 1, NULL, 1000018);

INSERT INTO attendances(att_date, att_status, att_remark, stu_id)
VALUES ('2020-12-24 17:00:00', 1, 'Late', 1000019);
```

```sql
INSERT INTO attendances(att_date, att_status, att_remark, stu_id)
VALUES ('2020-12-24 17:00:00', 1, 'Late', 1000020);
```

```sql
INSERT INTO enrollments(enroll_date, course_id, stu_id)
VALUES ('2019-09-25', 1, 1000001);

INSERT INTO enrollments(enroll_date, course_id, stu_id)
VALUES ('2019-09-26', 1, 1000002);

INSERT INTO enrollments(enroll_date, course_id, stu_id)
VALUES ('2019-09-25', 1, 1000003);

INSERT INTO enrollments(enroll_date, course_id, stu_id)
VALUES ('2019-09-26', 1, 1000004);

INSERT INTO enrollments(enroll_date, course_id, stu_id)
VALUES ('2019-09-28', 1, 1000005);

INSERT INTO enrollments(enroll_date, course_id, stu_id)
VALUES ('2019-09-28', 1, 1000006);

INSERT INTO enrollments(enroll_date, course_id, stu_id)
VALUES ('2019-09-28', 1, 1000007);

INSERT INTO enrollments(enroll_date, course_id, stu_id)
VALUES ('2019-10-01', 1, 1000008);

INSERT INTO enrollments(enroll_date, course_id, stu_id)
VALUES ('2019-10-03', 1, 1000009);

INSERT INTO enrollments(enroll_date, course_id, stu_id)
VALUES ('2019-10-06', 1, 1000010);

INSERT INTO enrollments(enroll_date, course_id, stu_id)
VALUES ('2019-10-06', 1, 1000011);

INSERT INTO enrollments(enroll_date, course_id, stu_id)
VALUES ('2019-10-07', 1, 1000012);

INSERT INTO enrollments(enroll_date, course_id, stu_id)
VALUES ('2019-10-10', 1, 1000013);

INSERT INTO enrollments(enroll_date, course_id, stu_id)
VALUES ('2019-10-12', 1, 1000014);

INSERT INTO enrollments(enroll_date, course_id, stu_id)
VALUES ('2019-10-13', 1, 1000015);

INSERT INTO enrollments(enroll_date, course_id, stu_id)
VALUES ('2019-10-13', 1, 1000016);

INSERT INTO enrollments(enroll_date, course_id, stu_id)
VALUES ('2019-10-17', 1, 1000017);

INSERT INTO enrollments(enroll_date, course_id, stu_id)
VALUES ('2019-10-18', 1, 1000018);

INSERT INTO enrollments(enroll_date, course_id, stu_id)
VALUES ('2019-10-25', 1, 1000019);
```

```
INSERT INTO enrollments(enroll_date, course_id, stu_id)
VALUES ('2019-11-01', 1, 1000020);
```

```sql
INSERT INTO subjects(sub_name, sub_credit, sub_startdate, sub_enddate, course_id)
VALUES ('C Programming 1', 25, '2019-11-06', '2020-03-05', 1);

INSERT INTO subjects(sub_name, sub_credit, sub_startdate, sub_enddate, course_id)
VALUES ('Fundamental Computer 1', 25, '2019-11-09', '2020-03-04', 1);

INSERT INTO subjects(sub_name, sub_credit, sub_startdate, sub_enddate, course_id)
VALUES ('Mathematics 1', 25, '2019-11-07', '2020-03-05', 1);

INSERT INTO subjects(sub_name, sub_credit, sub_startdate, sub_enddate, course_id)
VALUES ('Electronic 1', 25, '2019-11-07', '2020-03-07', 1);

INSERT INTO subjects(sub_name, sub_credit, sub_startdate, sub_enddate, course_id)
VALUES ('21st Century', 25, '2019-11-06', '2020-03-04', 1);

INSERT INTO subjects(sub_name, sub_credit, sub_startdate, sub_enddate, course_id)
VALUES ('English for Computer 1', 25, '2019-11-08', '2020-03-08', 1);

INSERT INTO subjects(sub_name, sub_credit, sub_startdate, sub_enddate, course_id)
VALUES ('C Programming 2', 25, '2020-07-28', '2020-11-03', 1);

INSERT INTO subjects(sub_name, sub_credit, sub_startdate, sub_enddate, course_id)
VALUES ('Fundamental Computer 2', 25, '2020-07-27', '2020-11-02', 1);

INSERT INTO subjects(sub_name, sub_credit, sub_startdate, sub_enddate, course_id)
VALUES ('Electronic 2', 25, '2020-07-30', '2020-11-05', 1);

INSERT INTO subjects(sub_name, sub_credit, sub_startdate, sub_enddate, course_id)
VALUES ('Khmer civilization', 25, '2020-07-29', '2020-11-04', 1);

INSERT INTO subjects(sub_name, sub_credit, sub_startdate, sub_enddate, course_id)
VALUES ('History', 25, '2020-07-27', '2020-11-02', 1);

INSERT INTO subjects(sub_name, sub_credit, sub_startdate, sub_enddate, course_id)
VALUES ('Mathematic 2', 25, '2020-07-28', '2020-11-03', 1);

INSERT INTO subjects(sub_name, sub_credit, sub_startdate, sub_enddate, course_id)
VALUES ('English 2', 25, '2020-07-27', '2020-11-06', 1);

INSERT INTO subjects(sub_name, sub_credit, sub_startdate, sub_enddate, course_id)
VALUES ('Database System 1', 25, '2020-12-24', '2021-04-22', 1);

INSERT INTO subjects(sub_name, sub_credit, sub_startdate, sub_enddate, course_id)
VALUES ('C++ Programming 1', 25, '2020-12-24', '2021-04-22', 1);

INSERT INTO subjects(sub_name, sub_credit, sub_startdate, sub_enddate, course_id)
VALUES ('Computer Architecture 1', 25, '2020-12-21', '2021-04-19', 1);

INSERT INTO subjects(sub_name, sub_credit, sub_startdate, sub_enddate, course_id)
VALUES ('Data structure 1', 25, '2020-12-23', '2021-04-21', 1);

INSERT INTO subjects(sub_name, sub_credit, sub_startdate, sub_enddate, course_id)
VALUES ('Data communication 1', 25, '2020-12-21', '2021-04-19', 1);

INSERT INTO subjects(sub_name, sub_credit, sub_startdate, sub_enddate, course_id)
VALUES ('English 3', 25, '2020-12-22', '2021-04-20', 1);
```

```
INSERT INTO subjects(sub_name, sub_credit, sub_startdate, sub_enddate, course_id)
VALUES ('Statistic Math 1', 25, '2019-11-06', '2020-03-05', 2);
```

```sql
INSERT INTO grades(grade_score, grade_year, grade_semester, sub_id, stu_id)
VALUES (90, 2, 1, 14, 1000001);

INSERT INTO grades(grade_score, grade_year, grade_semester, sub_id, stu_id)
VALUES (73, 2, 1, 14, 1000002);

INSERT INTO grades(grade_score, grade_year, grade_semester, sub_id, stu_id)
VALUES (92, 2, 1, 14, 1000003);

INSERT INTO grades(grade_score, grade_year, grade_semester, sub_id, stu_id)
VALUES (86, 2, 1, 14, 1000004);

INSERT INTO grades(grade_score, grade_year, grade_semester, sub_id, stu_id)
VALUES (95, 2, 1, 14, 1000005);

INSERT INTO grades(grade_score, grade_year, grade_semester, sub_id, stu_id)
VALUES (81, 2, 1, 14, 1000006);

INSERT INTO grades(grade_score, grade_year, grade_semester, sub_id, stu_id)
VALUES (82, 2, 1, 14, 1000007);

INSERT INTO grades(grade_score, grade_year, grade_semester, sub_id, stu_id)
VALUES (74, 2, 1, 14, 1000008);

INSERT INTO grades(grade_score, grade_year, grade_semester, sub_id, stu_id)
VALUES (81, 2, 1, 14, 1000009);

INSERT INTO grades(grade_score, grade_year, grade_semester, sub_id, stu_id)
VALUES (96, 2, 1, 14, 1000010);

INSERT INTO grades(grade_score, grade_year, grade_semester, sub_id, stu_id)
VALUES (88, 2, 1, 14, 1000011);

INSERT INTO grades(grade_score, grade_year, grade_semester, sub_id, stu_id)
VALUES (90, 2, 1, 14, 1000012);

INSERT INTO grades(grade_score, grade_year, grade_semester, sub_id, stu_id)
VALUES (87, 2, 1, 14, 1000013);

INSERT INTO grades(grade_score, grade_year, grade_semester, sub_id, stu_id)
VALUES (70, 2, 1, 14, 1000014);

INSERT INTO grades(grade_score, grade_year, grade_semester, sub_id, stu_id)
VALUES (90, 2, 1, 14, 1000015);

INSERT INTO grades(grade_score, grade_year, grade_semester, sub_id, stu_id)
VALUES (87, 2, 1, 14, 1000016);

INSERT INTO grades(grade_score, grade_year, grade_semester, sub_id, stu_id)
VALUES (92, 2, 1, 14, 1000017);

INSERT INTO grades(grade_score, grade_year, grade_semester, sub_id, stu_id)
VALUES (73, 2, 1, 14, 1000018);

INSERT INTO grades(grade_score, grade_year, grade_semester, sub_id, stu_id)
VALUES (85, 2, 1, 14, 1000019);
```

```
INSERT INTO grades(grade_score, grade_year, grade_semester, sub_id, stu_id)
VALUES (90, 2, 1, 14, 1000020);
```

```sql
INSERT INTO teachers(teach_name, teach_gender, teach_bd, teach_phone, teach_address,
teach_status)
VALUES ('Hun Sroy', 'M', '1988-01-28', '010251699', 'Phnom Penh', 1);

INSERT INTO teachers(teach_name, teach_gender, teach_bd, teach_phone, teach_address,
teach_status)
VALUES ('Kan Leang', 'M', '1979-02-19', '016125899', 'Kampong Cham', 1);

INSERT INTO teachers(teach_name, teach_gender, teach_bd, teach_phone, teach_address,
teach_status)
VALUES ('So Pheak', 'M', '1995-03-13', '092888799', 'Phnom Penh', 1);

INSERT INTO teachers(teach_name, teach_gender, teach_bd, teach_phone, teach_address,
teach_status)
VALUES ('Thea Srun', 'M', '1978-04-18', '08828899', 'Battambang', 1);

INSERT INTO teachers(teach_name, teach_gender, teach_bd, teach_phone, teach_address,
teach_status)
VALUES ('Vy Thida', 'F', '1981-05-06', '011231892', 'Banteay Meanchey', 1);

INSERT INTO teachers(teach_name, teach_gender, teach_bd, teach_phone, teach_address,
teach_status)
VALUES ('Dy Mangky', 'M', '1988-06-09', '012888859', 'Phnom Penh', 1);

INSERT INTO teachers(teach_name, teach_gender, teach_bd, teach_phone, teach_address,
teach_status)
VALUES ('Chan Dara', 'M', '1985-02-12', '012356865', 'Phnom Penh', 1);

INSERT INTO teachers(teach_name, teach_gender, teach_bd, teach_phone, teach_address,
teach_status)
VALUES ('Chhom Davit', 'M', '1987-12-30', '015522525', 'Phnom Penh', 1);

INSERT INTO teachers(teach_name, teach_gender, teach_bd, teach_phone, teach_address,
teach_status)
VALUES ('Thy Danuth', 'M', '1991-08-12', '012250654', 'Prey Veng', 1);

INSERT INTO teachers(teach_name, teach_gender, teach_bd, teach_phone, teach_address,
teach_status)
VALUES ('Kong Lida', 'F', '1973-05-01', '015580425', 'Phnom Penh', 1);

INSERT INTO teachers(teach_name, teach_gender, teach_bd, teach_phone, teach_address,
teach_status)
VALUES ('Long Rathana', 'M', '1899-02-20', '011522501', 'Phnom Penh', 1);

INSERT INTO teachers(teach_name, teach_gender, teach_bd, teach_phone, teach_address,
teach_status)
VALUES ('Lin Darith', 'M', '1990-02-20', '010020302', 'Phnom Penh', 1);

INSERT INTO teachers(teach_name, teach_gender, teach_bd, teach_phone, teach_address,
teach_status)
VALUES ('Kin Dina', 'F', '1994-04-30', '098474521', 'Phnom Penh', 1);

INSERT INTO teachers(teach_name, teach_gender, teach_bd, teach_phone, teach_address,
teach_status)
VALUES ('Lin Naroth', 'M', '1994-12-06', '098461521', 'Phnom Penh', 1);
```

```sql
INSERT INTO teachers(teach_name, teach_gender, teach_bd, teach_phone, teach_address,
teach_status)
VALUES ('Yin Vanuth', 'M', '1989-04-04', '012580656', 'Kampong Cham', 1);

INSERT INTO teachers(teach_name, teach_gender, teach_bd, teach_phone, teach_address,
teach_status)
VALUES ('Dy Rotha', 'F', '1991-02-06', '093543524', 'Phnom Penh', 1);

INSERT INTO teachers(teach_name, teach_gender, teach_bd, teach_phone, teach_address,
teach_status)
VALUES ('Yuth Chivon', 'M', '1978-08-25', '012823871', 'Takav', 1);

INSERT INTO teachers(teach_name, teach_gender, teach_bd, teach_phone, teach_address,
teach_status)
VALUES ('Hong Vitou', 'M', '1987-12-30', '016557854', 'Battambang', 1);

INSERT INTO teachers(teach_name, teach_gender, teach_bd, teach_phone, teach_address,
teach_status)
VALUES ('Chea Davin', 'M', '1977-04-11', '012542487', 'Phnom Penh', 1);

INSERT INTO teachers(teach_name, teach_gender, teach_bd, teach_phone, teach_address,
teach_status)
VALUES ('Chak Riya', 'F', '1992-12-05', '092518792', 'Phnom Penh', 1);
```

```
INSERT INTO instructors(teach_id, sub_id)
VALUES (1000006, 14);

INSERT INTO instructors(teach_id, sub_id)
VALUES (1000007, 14);

INSERT INTO instructors(teach_id, sub_id)
VALUES (1000008, 14);

INSERT INTO instructors(teach_id, sub_id)
VALUES (1000003, 15);

INSERT INTO instructors(teach_id, sub_id)
VALUES (1000002, 15);

INSERT INTO instructors(teach_id, sub_id)
VALUES (1000004, 16);

INSERT INTO instructors(teach_id, sub_id)
VALUES (1000005, 16);

INSERT INTO instructors(teach_id, sub_id)
VALUES (1000001, 15);

INSERT INTO instructors(teach_id, sub_id)
VALUES (1000017, 17);

INSERT INTO instructors(teach_id, sub_id)
VALUES (1000018, 18);

INSERT INTO instructors(teach_id, sub_id)
VALUES (1000019, 18);

INSERT INTO instructors(teach_id, sub_id)
VALUES (1000015, 17);

INSERT INTO instructors(teach_id, sub_id)
VALUES (1000014, 19);

INSERT INTO instructors(teach_id, sub_id)
VALUES (1000013, 19);

INSERT INTO instructors(teach_id, sub_id)
VALUES (1000016, 17);

INSERT INTO instructors(teach_id, sub_id)
VALUES (1000009, 16);

INSERT INTO instructors(teach_id, sub_id)
VALUES (1000020, 18);

INSERT INTO instructors(teach_id, sub_id)
VALUES (1000012, 19);

INSERT INTO instructors(teach_id, sub_id)
VALUES (1000010, 1);
```

```
INSERT INTO instructors(teach_id, sub_id)
VALUES (1000011, 1);
```