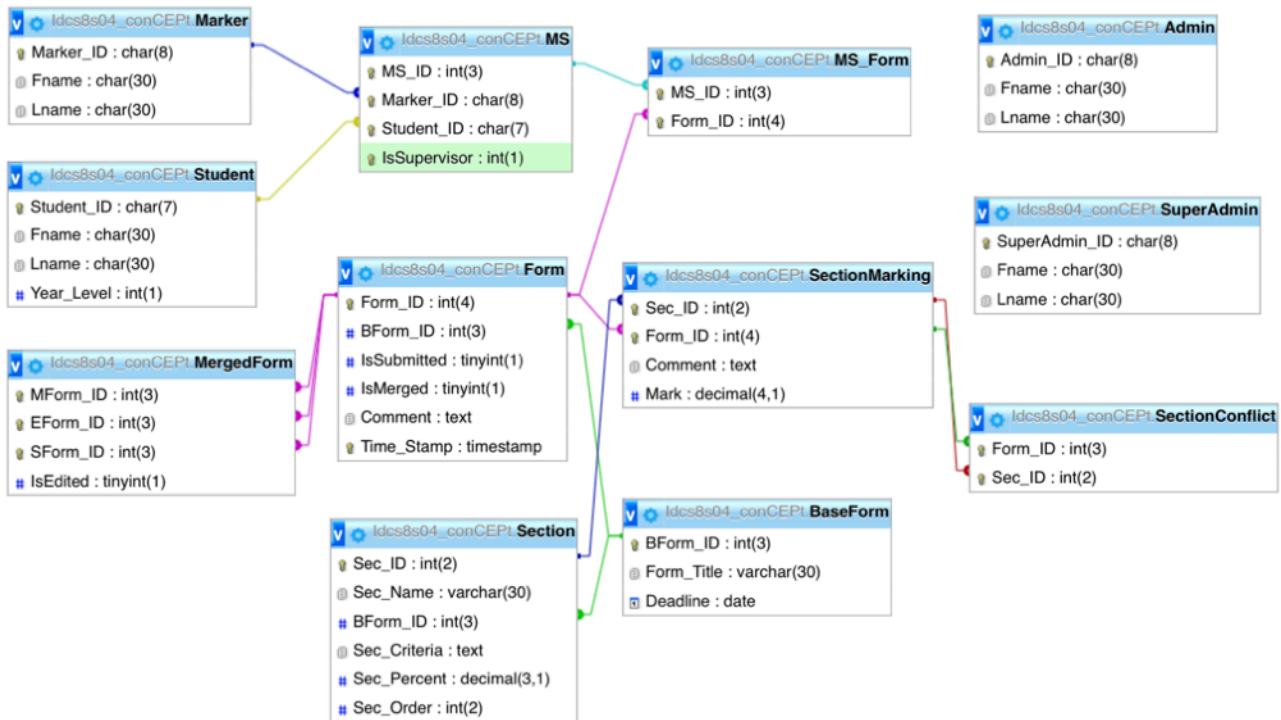


# Database Guide

This document will endeavour to explain the purpose of the conCEPT database's many elements and how to query it. <sup>1</sup>

## Overall Design



## Tables

### Marker and Student

Use the markers' and students' ids as primary keys. This will guarantee that each entry is unique and would help with dealing with duplicated names. Also, all the needed information such as name can be retrieved from the university server.

### MS

Markers' are linked to the students. An integer flag is set to determine if the marker is a supervisor (1) or an examiner (0).

### BaseForm

This table helps in creating a general template for the forms. It contains the title of the main forms, e.g. Design Report, and possibly a deadline that the forms need to be submitted (this field is not implemented in the system but can be added if desired). Only the super admin can edit anything in this table.

### Section

This table helps in creating a general template for the sections. It contains all the main information of a section, e.g. criteria, name, and weighting. Each section is linked to a base form.

---

<sup>1</sup> All foreign keys cascade on delete and update

## Form

All the forms created are saved in this table. Each form has a comment and a time stamp associated with it. This is to keep track of previous submission of a form. Also, two integer flags are included to determine if a form has been submitted (1) or not (0) and if a form is a merged form (-1), has been merged or part of a merged form (1), or neither (0). Each form is linked to a base form and by such can be linked to the forms' sections.

## MS\_Form

Each marker-student pair (MS) are linked to at least 5 forms - the base forms. The marker-student can be linked to one or more form of the same base form (previous submissions of the forms are saved). Merged forms won't be linked to marker-student pair in this table, refer to MergedForm table.

## SectionMarking

Sections are linked to a form which shares the same base form. The section and form ids are the primary keys of this table and are also foreign keys. The comments and marks of a student are stored in this table.

## MergedForm (special case)

The supervisor and examiner forms are linked to the merged form. The ids of each form is stored in the table along with an integer flag to indicate if a merged form has been edited (1) or not (0).

## SectionConflict

When the supervisor and examiner forms are merged, the marks for each section are compared. If the difference between the mark of a section is more than 10%, then that section and the related merged form are stored in this table.

## Admin

A table to keep track of the staff that can add and link students to markers as the admins of the system. Also, they can save the completed marked forms as pdfs. Similarly to the marker's table, the user id and name of a staff are stored.

## SuperAdmin

Just like admin this table keeps track of the staff that have total control over the system e.g. editing or deleting a base form.

## Interacting with the system

### Queries to get information from the database<sup>2</sup>

(1) To get base form title, id and deadline

```
SELECT * FROM BaseForm WHERE 1
```

(2) To get general form details for blank form - based on base form id

```
SELECT Sec_Name , Sec_Criteria , Sec_Percent , Sec_Order
FROM Section
WHERE Section.BForm_ID = 1
```

(3) To get a specific form information - based on form ID

```
SELECT Section.Sec_Order , Section.Sec_Name , Section.Sec_Percent ,
Section.Sec_Criteria , SectionMarking.Comment , SectionMarking.Mark
```

---

<sup>2</sup> Most queries depend heavily on the form id

```
FROM SectionMarking
JOIN Section ON Section.Sec_ID = SectionMarking.Sec_ID
WHERE SectionMarking.Form_ID = 1
```

**(4) To get section weighting and mark of a form - based on form ID**

```
SELECT Section.Sec_Percent , SectionMarking.Mark
FROM SectionMarking
JOIN Section ON Section.Sec_ID = SectionMarking.Sec_ID
WHERE SectionMarking.Form_ID = 1
```

**(5) To get total mark of a form - based on form ID**

```
SELECT SUM(Section.Sec_Percent*SectionMarking.Mark / 100) AS Total
FROM SectionMarking
JOIN Section ON Section.Sec_ID = SectionMarking.Sec_ID
WHERE SectionMarking.Form_ID = 1
```

**(6) To get student name and year for a specific form - based on form ID**

```
SELECT Student.Fname , Student.Lname , Student.Year_Level
FROM MS_Form
JOIN MS ON MS.MS_ID = MS_Form.MS_ID
JOIN Student ON Student.Student_ID = MS.Student_ID
WHERE MS_Form.Form_ID = 1
```

**(7) To get markers for a specific form - based on form ID**

```
SELECT Marker.Fname , Marker.Lname , MS.IsSupervisor
FROM MS_Form
JOIN MS ON MS.MS_ID = MS_Form.MS_ID
JOIN Marker ON Marker.Marker_ID = MS.Marker_ID
WHERE MS_Form.Form_ID = 1
```

**(8) To get form title and is submitted or is merged - based on form ID**

```
SELECT BaseForm.Form_title , Form.IsSubmitted , Form.IsMerged
FROM Form
JOIN BaseForm ON BaseForm.BForm_ID = Form.BForm_ID
WHERE Form.Form_ID = 1
```

**(9) To get students and their forms for a specific marker - based on marker ID**

```
SELECT Form.Form_ID, BaseForm.Form_title, Student.Fname , Student.Lname ,
Student.Year_Level
FROM MS_Form
JOIN MS ON MS.MS_ID = MS_Form.MS_ID
```

```

JOIN Marker ON Marker.Marker_ID = MS.Marker_ID
JOIN Student ON Student.Student_ID = MS.Student_ID
JOIN Form ON Form.Form_ID = MS_Form.Form_ID
JOIN BaseForm ON BaseForm.BForm_ID = Form.BForm_ID
WHERE Marker.Marker_ID = 'hkd4hdk'

```

**(10)To get students and their forms for a specific marker where the marker is a supervisor or not - based on marker ID and isSupervisor flag (optional)**

```

SELECT Form.Form_ID, BaseForm.Form_title, Student.Fname , Student.Lname ,
Student.Year_Level
FROM MS_Form
JOIN MS ON MS.MS_ID = MS_Form.MS_ID
JOIN Marker ON Marker.Marker_ID = MS.Marker_ID
JOIN Student ON Student.Student_ID = MS.Student_ID
JOIN Form ON Form.Form_ID = MS_Form.Form_ID
JOIN BaseForm ON BaseForm.BForm_ID = Form.BForm_ID
WHERE Marker.Marker_ID = 'hkd4hdk' AND MS.IsSupervisor = 1

```

**(11)To get students for a specific base form - based on marker id and base form id**

```

SELECT Form.Form_ID, Student.Fname , Student.Lname , Student.Year_Level
FROM MS_Form
JOIN MS ON MS.MS_ID = MS_Form.MS_ID
JOIN Marker ON Marker.Marker_ID = MS.Marker_ID
JOIN Student ON Student.Student_ID = MS.Student_ID
JOIN Form ON Form.Form_ID = MS_Form.Form_ID
JOIN BaseForm ON BaseForm.BForm_ID = Form.BForm_ID
WHERE Marker.Marker_ID = 'hkd4hdk' AND BaseForm.BForm_ID = 1

```

**(12)To get markers for a merged form - based on merged from id**

```

SELECT Marker.Fname, Marker.Lname
FROM MergedForm
JOIN MS_Form ON MS_Form.Form_ID = MergedForm.EForm_ID OR MS_Form.Form_ID
= MergedForm.SForm_ID
JOIN MS ON MS.MS_ID = MS_Form.MS_ID
JOIN Marker ON Marker.Marker_ID = MS.Marker_ID
WHERE MergedForm.MForm_ID = 8

```

**(13)To get student for a merged form - based on merged from id**

```

SELECT Student.Fname , Student.Lname , Student.Year_Level
FROM MergedForm

```

```

JOIN MS_Form ON MS_Form.Form_ID = MergedForm.EForm_ID
JOIN MS ON MS.MS_ID = MS_Form.MS_ID
JOIN Student ON Student.Student_ID = MS.Student_ID
WHERE MergedForm.MForm_ID = 9

```

**(14)To get base form id, student id and if a marker is supervisor - based on form id**

```

SELECT Form.BForm_ID, Student.Student_ID, MS.IsSupervisor
FROM Form
JOIN MS_Form ON MS_Form.Form_ID = Form.Form_ID
JOIN MS ON MS.MS_ID = MS_Form.MS_ID
JOIN Student ON Student.Student_ID = MS.Student_ID
WHERE Form.Form_ID = 1

```

**(15)To get the form ID of the other marker if he has submitted his form - based on student id, base form id, form id, and isSupervisor**

```

SELECT Form.Form_ID
FROM Form
JOIN MS_Form ON MS_Form.Form_ID = Form.Form_ID
JOIN MS ON MS.MS_ID = MS_Form.MS_ID
JOIN Student ON Student.Student_ID = MS.Student_ID
WHERE Student.Student_ID = 'cnng04' AND Form.BForm_ID = 1
AND Form.Form_ID != 1 AND Form.IsSubmitted = 1 AND MS.IsSupervisor != 1
ORDER BY Form.Time_Stamp DESC

```

**(16)To get the merged form id - based on a form id that is part of the merged form**

```

SELECT MForm_ID
FROM MergedForm
WHERE EForm_ID = 1 OR SForm_ID = 1

```

**(17)To get the sections causing conflict in a merged form - based on section form ID**

```

SELECT Sec_ID
FROM SectionConflict
WHERE Form_ID = 8

```

**(18>Returns 1 if the current marker is the supervisor for this merged form, 0 if examiner - based on merged form id and marker id**

```

SELECT MS.IsSupervisor
FROM MergedForm
JOIN MS_Form ON MS_Form.Form_ID = MergedForm.EForm_ID OR MS_Form.Form_ID
= MergedForm.SForm_ID
JOIN MS ON MS.MS_ID = MS_Form.MS_ID

```

```
WHERE MergedForm.MForm_ID = 8 AND MS.Marker_ID = 'hkd4hdk'
```

**(19) To check if a form is edited - based on merged form id**

```
SELECT IsEdited  
FROM MergedForm  
WHERE MForm_ID = 8
```

**(20) To find un-used merged form id - based on base form id<sup>3</sup>**

```
SELECT Form_ID  
FROM Form  
LEFT JOIN MergedForm ON MergedForm.MForm_ID = Form.Form_ID  
WHERE IsMerged = -1 AND MergedForm.MForm_ID IS NULL AND BForm_ID = 1  
ORDER BY Form_ID ASC  
LIMIT 1
```

**(21) To get all completed forms sorted by studentID**

```
SELECT Student.Student_ID, Student.Fname, Student.Lname, Form.Form_ID,  
BaseForm.Form_Title  
FROM Form  
JOIN BaseForm ON BaseForm.BForm_ID = Form.BForm_ID  
JOIN MergedForm ON MergedForm.MForm_ID = Form.Form_ID  
JOIN MS_Form ON MS_Form.Form_ID = MergedForm.EForm_ID  
JOIN MS ON MS.MS_ID = MS_Form.MS_ID  
JOIN Student ON Student.Student_ID = MS.Student_ID  
WHERE Form.IsSubmitted = 1 AND Form.IsMerged = -1  
ORDER BY Student.Student_ID ASC
```

## **Queries for adding data to the database**

**(1) To insert a comment and mark of a specific section - based on its order and the form ID**

```
INSERT INTO SectionMarking(Sec_ID, Form_ID, Comment, Mark)  
SELECT Section.Sec_ID, 1, 'bad', 50.0  
FROM Section  
JOIN Form ON Form.BForm_ID = Section.BForm_ID  
WHERE Form.Form_ID = 1 AND Section.Sec_Order = 1
```

--> for inserting multiple values at once

```
INSERT INTO SectionMarking(Sec_ID, Form_ID, Comment, Mark)  
SELECT Section.Sec_ID, 2, 'bad', 50.0
```

---

<sup>3</sup> This query is used to find the new merged form that has been inserted to the form table

```

FROM Section
JOIN Form ON Form.BForm_ID = Section.BForm_ID
WHERE Form.Form_ID = 2 AND Section.Sec_Order = 1
UNION ALL
SELECT Section.Sec_ID, 2 , 'GOOD', 60.0
FROM Section
JOIN Form ON Form.BForm_ID = Section.BForm_ID
WHERE Form.Form_ID = 2 AND Section.Sec_Order = 2

```

**(2) To update a section - based on the section order and form ID**

```

UPDATE SectionMarking
JOIN Section ON Section.Sec_ID = SectionMarking.Sec_ID
SET SectionMarking.Comment = 'Good job!',SectionMarking.Mark= 80
WHERE SectionMarking.Form_ID = 1 AND Section.Sec_Order = 1

```

**(3) To update submit form flag - based on form id**

```

UPDATE Form
SET IsSubmitted= 1, Time_Stamp = NOW()
WHERE Form_ID = 1

```

**(4) To insert a new merged form - based on examiner form id and supervisor form id**

```

INSERT INTO Form(BForm_ID, IsMerged)
SELECT Form.BForm_ID, -1
FROM Form
WHERE Form.Form_ID = 3;

```

```

INSERT INTO MergedForm(MForm_ID,EForm_ID,SForm_ID)
SELECT LAST_INSERT_ID(), 3, 1;

```

```

SET group_concat_max_len = 65535;

```

```

INSERT INTO SectionMarking(Sec_ID, Form_ID, Comment, Mark)
SELECT Sec_ID, LAST_INSERT_ID() , GROUP_CONCAT(Comment SEPARATOR ' '),
AVG(Mark)
FROM (
SELECT Sec_ID, Form_ID, Comment, Mark FROM SectionMarking WHERE Form_ID = 3
UNION ALL
SELECT Sec_ID, Form_ID, Comment, Mark FROM SectionMarking WHERE Form_ID =

```

) AS B

GROUP BY Sec\_ID

**(5) To insert conflicted section into the sectionConflict table - based on merged form id and an examiner or supervisor form id**

```
INSERT INTO SectionConflict(Form_ID,Sec_ID)
```

```
SELECT s1.Form_ID,s1.Sec_ID
```

```
FROM (SELECT Sec_ID, Form_ID, Comment, Mark FROM SectionMarking WHERE  
Form_ID = 39) s1
```

```
INNER JOIN (SELECT Sec_ID, Form_ID, Comment, Mark FROM SectionMarking WHERE  
Form_ID = 1) s2 ON s2.Sec_ID = s1.Sec_ID
```

```
WHERE s1.Form_ID = 39 AND s2.Form_ID= 3 AND s1.Mark - s2.Mark > 5 OR s2.Mark -  
s1.Mark > 5
```

**(6) To duplicate a form - based on form id**

```
INSERT INTO Form(BForm_ID, IsSubmitted, IsMerged, Comment, Time_Stamp)
```

```
SELECT BForm_ID, IsSubmitted, IsMerged, Comment, Time_Stamp
```

```
FROM Form
```

```
WHERE Form_ID = 3;
```

```
INSERT INTO MS_Form(MS_ID, Form_ID)
```

```
SELECT MS_ID, LAST_INSERT_ID()
```

```
FROM MS_Form
```

```
WHERE Form_ID = 3;
```

```
INSERT INTO SectionMarking(Sec_ID, Form_ID, Comment, Mark)
```

```
SELECT Sec_ID, LAST_INSERT_ID() , Comment, Mark
```

```
FROM (
```

```
SELECT Sec_ID, Form_ID, Comment, Mark FROM SectionMarking WHERE Form_ID = 3
```

```
) AS B
```

```
GROUP BY Sec_ID
```

**(7) To update/add a merged form information in sectionMarking table - based on merged form id**

```
INSERT INTO SectionMarking(Sec_ID, Form_ID, Comment, Mark)
```

```
SELECT Sec_ID, 39 , GROUP_CONCAT(Comment SEPARATOR ' '), AVG(Mark)
```

```
FROM (
```

```
SELECT Sec_ID, Form_ID, Comment, Mark FROM SectionMarking WHERE Form_ID = 3
```

```
UNION ALL
```



```

        SELECT Sec_ID, Form_ID, Comment, Mark FROM SectionMarking WHERE Form_ID =
1
    ) AS B
GROUP BY Sec_ID
ON DUPLICATE KEY UPDATE
    Sec_ID = VALUES(Sec_ID), Comment = VALUES(Comment), Mark= VALUES(Comment)

```

**(8) To add/update a student**

```

INSERT INTO Student(Student_ID, Fname, Lname, Year_Level)
VALUES ('knlb5j','Luke','McBeth',3)
ON DUPLICATE KEY UPDATE
    Fname = VALUES(Fname), Lname = VALUES(Lname), Year_Level= VALUES(Year_Level)

```

**(9) To add/update marker**

```

INSERT INTO Marker(Marker_ID, Fname, Lname)
VALUES ('lmlhi6','Mark','Million')
ON DUPLICATE KEY UPDATE
    Fname = VALUES(Fname), Lname = VALUES(Lname)

```

**(10)To connect marker and student - based on marker and student id**

```

INSERT INTO MS(Marker_ID, Student_ID, IsSupervisor)
VALUES ('lmlhi6','knlb5j',1)

```

**(11)To create a new form for each base (without linking the sections)<sup>4</sup>**

```

INSERT INTO Form(BForm_ID)
SELECT BForm_ID
FROM BaseForm

```

**(12)To connect the marker-student pair to the 5 form types (forms previously created but not used) - based on student and marker id**

```

INSERT INTO MS_Form(MS_ID, Form_ID)
SELECT
(
    SELECT MS.MS_ID
    FROM MS
    LEFT JOIN MS_Form ON MS_Form.MS_ID = MS.MS_ID
    WHERE MS_Form.MS_ID IS NULL AND MS.Marker_ID = 'lmlhi6' AND MS.Student_ID =
'knlb5j'
) M , Form_ID
FROM

```

---

<sup>4</sup> Auto-increment skips values when it runs this query.

```
(
    SELECT Form.Form_ID, Form.BForm_ID
FROM Form
LEFT JOIN MS_Form ON MS_Form.Form_ID = Form.Form_ID
WHERE MS_Form.Form_ID IS NULL AND IsMerged = 0
    GROUP BY Form.BForm_ID
) B
```

**(13) To add the sections to the forms created for the marker-student pair - base on marker and student ids**

```
INSERT INTO SectionMarking(Sec_ID, Form_ID)
SELECT Section.Sec_ID, Form.Form_ID
FROM Form
JOIN Section ON Section.BForm_ID = Form.BForm_ID
JOIN MS_Form ON MS_Form.Form_ID = Form.Form_ID
JOIN MS ON MS.MS_ID = MS_Form.MS_ID
WHERE MS.Marker_ID = 'lmlhi6' AND MS.Student_ID = 'knlb5j'
```

**(14) To update criteria of a form - based on base form id and section id**

```
UPDATE `Section`
SET `Sec_Name`=[value-2], `Sec_Criteria`=[value-4], `Sec_Percent`=[value-5]
WHERE `BForm_ID` = 1 AND `Sec_Order`= 1
```

## Combining the Queries

♦ **To generate a new link between a student and a marker, then generate and link all forms to the new pair - need student and marker ids (Queries 10, 11, 12, and 13 from adding section)**

```
INSERT INTO MS(Marker_ID, Student_ID, IsSupervisor)
VALUES ('lmlhi6', 'knlb5j', 1);
```

```
INSERT INTO Form(BForm_ID)
SELECT BForm_ID
FROM BaseForm;
```

```
INSERT INTO MS_Form(MS_ID, Form_ID)
SELECT
(
    SELECT MS.MS_ID
```

```

FROM MS
LEFT JOIN MS_Form ON MS_Form.MS_ID = MS.MS_ID
WHERE MS_Form.MS_ID IS NULL AND MS.Marker_ID = 'lmlhi6' AND MS.Student_ID =
'knlb5j'
) M , Form_ID
FROM
(
    SELECT Form.Form_ID, Form.BForm_ID
FROM Form
LEFT JOIN MS_Form ON MS_Form.Form_ID = Form.Form_ID
WHERE MS_Form.Form_ID IS NULL AND IsMerged = 0
    GROUP BY Form.BForm_ID
) B;

```

```

INSERT INTO SectionMarking(Sec_ID, Form_ID)
SELECT Section.Sec_ID, Form.Form_ID
FROM Form
JOIN Section ON Section.BForm_ID = Form.BForm_ID
JOIN MS_Form ON MS_Form.Form_ID = Form.Form_ID
JOIN MS ON MS.MS_ID = MS_Form.MS_ID
WHERE MS.Marker_ID = 'lmlhi6' AND MS.Student_ID = 'knlb5j';

```

- ◆ **To Link marker-student pair to a new generated base form and to generate all forms and link them to the new base form's sections - need base form id**

```

INSERT INTO Form(BForm_ID)
SELECT 6
FROM MS;

```

```

SET @row_numberM = 0;
SET @row_numberF = 0;

```

```

INSERT MS_Form (MS_ID, Form_ID)
SELECT MS_ID, Form_ID
FROM (SELECT MS_ID, (@row_numberM:=@row_numberM + 1) AS num
FROM MS) m
JOIN (SELECT Form_ID, (@row_numberF:=@row_numberF + 1) AS num

```

```
FROM Form WHERE BForm_Id = 6) f ON f.num = m.num;
```

```
INSERT INTO SectionMarking(Sec_ID, Form_ID)
```

```
SELECT Section.Sec_ID, MS_Form.Form_ID
```

```
FROM Form
```

```
JOIN Section ON Section.BForm_ID = Form.BForm_ID
```

```
JOIN MS_Form ON MS_Form.Form_ID = Form.Form_ID
```

```
WHERE Form.BForm_ID = 6;
```