/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**CSE532 -- Project 2 Report**

**Author(s): Minghui Lin (109557872)**

**Yishuo Wang (108533945)**

**Report Documentation**

I (or We, if working with a partner) pledge my (or our) honor that all parts

of this project were done by me (or us) alone and without collaboration with

anybody else.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

***The division of work：***

***Yishuo Wang:***

***1 Discuss the ER diagram and design it.***

***2 Insert all the data***

***3 Change the default schema to the object relational schema.***

***4.Query 1,2,4 and Query1.jsp, Query2.jsp, Query4.jsp. Query5 direct part.***

***5.Write the report***

***Minghui Lin:***

***1 Discuss the ER diagram and design it..***

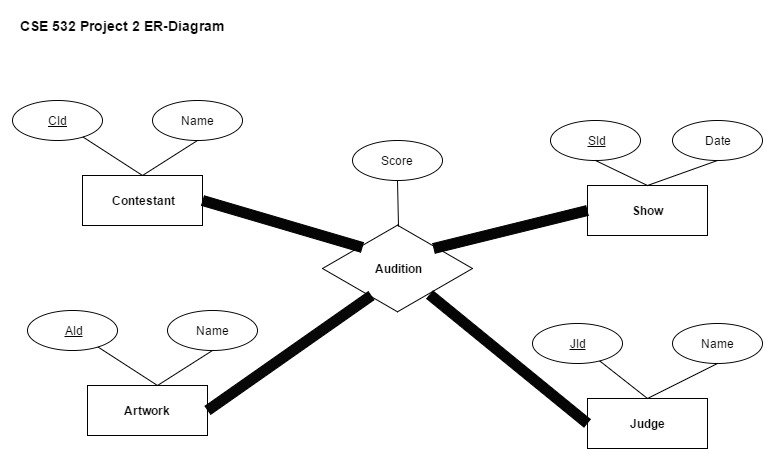
***2 Create the default schema.***

***3 Setup the jsp, jdbc, tomcat, Netbeans to run successfully.***

***4.Query 3, 5 and Query3.jsp, Query5.jsp***

***5.Write the report***

**Entity-Relationship (ER) diagram:**



**A clear description of the database scheme:**

1) There are 4 entity types and 1 relationship type in our database scheme.

2) The 4 entity types are: Contestant, Show, Artwork and Judge.

Attributes of Contestant are: CId and Name

Primary Key of Contestant: CId.

Attributes of Show are: SId and Date

Primary Key of Show: SId

Attributes of Artwork are: AId, Name

Primary Key of Artwork: AId

Attributes of Judge are: JId and Name

Primary Key of Judge: JId

The Audition is relationship type which connects to the entity types of Contestant, Show, Artwork and Judge.

Audition has its own attribute which is Score.

Audition has 4 foreign keys which are combined as its foreign key as well. They are:

CId, SId, AId and JId.

This database scheme is very simply, we can query information from Audition table and connect to the other entity type to get extra information if we needed.

**Description of integrity constraints, including referential integrity and CHECK-constraints whenever applicable:**

Contestant:

CId is Primary Key.

Show:

SId is Primary Key

Artwork:

AId is Primary Key

Judge:

JId is Primary Key

Audition:

CId, SId, AId and JId are foreign keys to CId in Contestant, SId in Show, AId in Artwork and JId in Judge.

CId, SId, AId and JId are Primary key of Audition.

**CHECK-constraints:**

We check Attribute of Score must be between 0 and 10.

**All SQL CREATE TABLE/VIEW/TYPE commands used to build the database. These statements must include all the applicable FOREIGN KEY and CHECK statements.**

**Create Types:**

CREATE TYPE public."Artwork\_Type" AS

(

"AId" character(30),

"Name" character(30)

);

CREATE TYPE public."Contestant\_Type" AS

(

"CId" character(30),

"Name" character(30)

);

CREATE TYPE public."Judge\_Type" AS

(

"JId" character(30),

"Name" character(30)

);

CREATE TYPE public."Show\_Type" AS

(

"SId" character(30),

"Date" date

);

CREATE TYPE public."Audition\_Type" AS

(

"Score" integer,

"CId" character(30),

"AId" character(30),

"SId" character(30),

"JId" character(30)

);

**Create Tables:**

CREATE TABLE public."Artwork"

OF "Artwork\_Type"

(

CONSTRAINT "Artwork\_pkey" PRIMARY KEY ("AId")

)

WITH (

OIDS = TRUE

);

CREATE TABLE public."Contestant"

OF "Contestant\_Type"

(

CONSTRAINT "Contestant\_pkey" PRIMARY KEY ("CId")

)

WITH (

OIDS = TRUE

);

CREATE TABLE public."Judge"

OF "Judge\_Type"

(

CONSTRAINT "Judge\_pkey" PRIMARY KEY ("JId")

)

WITH (

OIDS = TRUE

);

CREATE TABLE public."Show"

OF "Show\_Type"

(

CONSTRAINT "Show\_pkey" PRIMARY KEY ("SId")

)

WITH (

OIDS = TRUE

);

CREATE TABLE public."Audition"

OF "Audition\_Type"

(

CONSTRAINT "Audition\_pkey" PRIMARY KEY ("CId", "AId", "SId", "JId"),

CONSTRAINT "AId" FOREIGN KEY ("AId")

REFERENCES public."Artwork" ("AId") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION,

CONSTRAINT "CId" FOREIGN KEY ("CId")

REFERENCES public."Contestant" ("CId") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION,

CONSTRAINT "JId" FOREIGN KEY ("JId")

REFERENCES public."Judge" ("JId") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION,

CONSTRAINT "SId" FOREIGN KEY ("SId")

REFERENCES public."Show" ("SId") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION,

CONSTRAINT "Audition\_Score\_check" CHECK ("Score" >= 0 AND "Score" <= 10)

)

WITH (

OIDS = TRUE

);

**Create Views:**

We will create views based on different query, please see each query for different views we have created.

**For each query mentioned in Section 3, the report must supply the appropriate SQL statement.**

Query 1:

SELECT DISTINCT c1."Name", c2."Name"

FROM public."Contestant" c1, public."Contestant" c2,

public."Audition" a1, public."Audition" a2,

public."Show" s1, public."Show" s2

WHERE c1."CId" != c2."CId" AND c1."Name" < c2."Name"

AND a1."CId" = c1."CId" AND a2."CId" = c2."CId" AND a1."AId" = a2."AId" AND a1."Score" = a2."Score"

AND a1."SId" = s1."SId" AND a2."SId" = s2."SId" AND s1."Date" = s2."Date"

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

Query 2:

SELECT R.name1, R.name2 FROM

(

SELECT DISTINCT c1."Name" name1, c2."Name" name2,

MAX(a1."Score") OVER (PARTITION BY a1."CId", a1."SId", a1."AId") max1,

MAX(a2."Score") OVER (PARTITION BY a2."CId", a2."SId", a2."AId") max2,

MIN(a1."Score") OVER (PARTITION BY a1."CId", a1."SId", a1."AId") min1,

MIN(a2."Score") OVER (PARTITION BY a2."CId", a2."SId", a2."AId") min2

FROM "Contestant" c1, "Contestant" c2,

"Audition" a1, "Audition" a2

WHERE c1."CId" != c2."CId" AND c1."Name" < c2."Name"

AND a1."CId" = c1."CId" AND a2."CId" = c2."CId" AND a1."AId" = a2."AId"

GROUP BY name1, name2,

a1."Score", a2."Score",

a1."CId", a2."CId",

a1."SId", a2."SId",

a1."AId", a2."AId"

) R

WHERE R.max1 = R.max2 AND R.min1 = R.min2

ORDER BY R.name1

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

Query 3:

CREATE OR REPLACE VIEW "SAME\_ARTWORK" AS

SELECT a1."CId", a1."AId", count(a1."JId") "JUDGE\_NUM", avg(a1."Score") "AVG\_SCORE"

FROM "Audition" a1

GROUP BY a1."CId", a1."AId", a1."SId"

ORDER BY a1."CId";

SELECT DISTINCT C1."Name" name1, C2."Name" name2

FROM "SAME\_ARTWORK" S1, "SAME\_ARTWORK" S2, "Contestant" C1, "Contestant" C2

WHERE

S1."CId" = C1."CId" AND

S2."CId" = C2."CId" AND

S1."CId" != S2."CId" AND

C1."Name" < C2."Name" AND

S1."AId" = S2."AId" AND

S1."JUDGE\_NUM" = S2."JUDGE\_NUM" AND

S1."AVG\_SCORE" = S2."AVG\_SCORE"

ORDER BY C1."Name";

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

Query 4:

CREATE OR REPLACE VIEW "Show\_Contestants" AS

SELECT a1."CId", a1."SId"

FROM "Audition" a1

GROUP BY a1."CId", a1."SId"

ORDER BY a1."CId";

SELECT DISTINCT contestant1."Name", contestant2."Name"

FROM "Contestant" contestant1, "Contestant" contestant2

WHERE NOT EXISTS

(

SELECT DISTINCT c1."Name", c2."Name"

FROM

"Contestant" c1,

"Contestant" c2,

"Show\_Contestants" sc1,

"Show\_Contestants" sc2

WHERE

contestant1."CId" = c1."CId"

AND contestant2."CId" = c2."CId"

AND c1."CId" != c2."CId"

AND sc1."CId" = c1."CId"

AND sc2."CId" = c2."CId"

AND sc2."SId" NOT IN (SELECT sc3."SId"

FROM "Show\_Contestants" sc3

WHERE sc3."CId" = sc1."CId"

)

)

AND contestant1."CId" != contestant2."CId"

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

Query 5:

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

-- Recusive part(This is Recursive View) --

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

CREATE OR REPLACE RECURSIVE VIEW "CLOSRURE" (cid1, cid2) AS

SELECT DISTINCT view1."CId" cid1, view2."CId" cid2

FROM "AVG\_VIEW" view1, "AVG\_VIEW" view2,

"Audition" A1, "Audition" A2

WHERE

view1."CId" != view2."CId" AND

view1."CId" = A1."CId" AND

view2."CId" = A2."CId" AND

A1."AId" = A2."AId" AND

A1."SId" = A2."SId" AND

abs(view1."AVG\_SCORE" - view2."AVG\_SCORE") <= 0.2

UNION

SELECT DISTINCT CL."cid1" cid2, view2."CId" cid1

FROM "AVG\_VIEW" view1, "AVG\_VIEW" view2,

"Audition" A1, "Audition" A2,

"CLOSRURE" CL

WHERE

view1."CId" != view2."CId" AND

view1."CId" = A1."CId" AND

view2."CId" = A2."CId" AND

A1."AId" = A2."AId" AND

A1."SId" = A2."SId" AND

abs(view1."AVG\_SCORE" - view2."AVG\_SCORE") <= 0.2 AND

view1."CId" = CL."cid2";

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

-- Display final result part --

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

SELECT C1."Name", C2."Name"

FROM "CLOSRURE" CL, "Contestant" C1, "Contestant" C2

WHERE CL."cid1" = C1."CId" AND

CL."cid2" = C2."CId" AND

C1."Name" < C2."Name"

ORDER BY C1."Name",C2."Name";

**A brief user guide. Explain how to install and run your program.**

To Run our program is very simple.

Please open Netbean IDE and open our project under “Coding” directory and Press Run button in Netbean to run the program.

Please make sure you have installed Tomcat with NetBeans configuration.

The following screenshot are from our running program:

