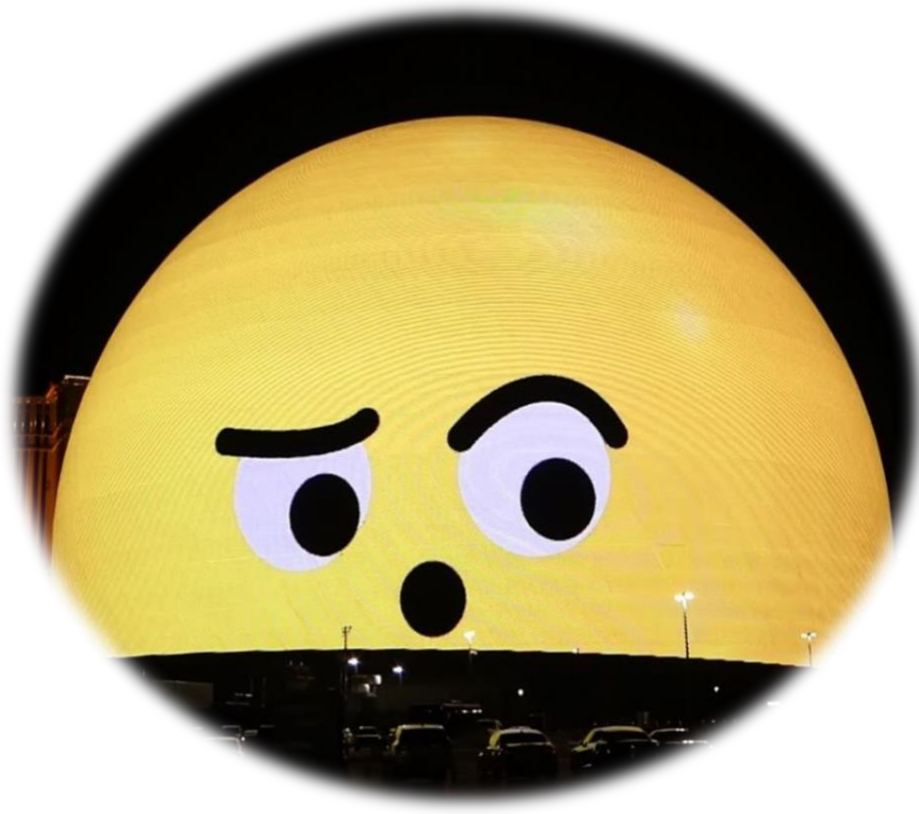


# Final Project

Database Management

~~CONFIDENTIAL~~



**Marist College**

School of Computer Science and Mathematics

Linda Mukundwa

[Linda.Mukundwa1@marist.edu](mailto:Linda.Mukundwa1@marist.edu)

~~CONFIDENTIAL~~

Fall 2023

# Table of Contents

<b>Business Rules</b>	
1. Overview of Rules	3
2. Entity Relationship Diagram (Chen)	5
<b>Tables</b>	
1. cAgency	6
2. cEntertainment	7
3. cVenue	8
4. cEmploys	9
5. cVendor	10
6. cRecruits	11
7. cEmployees	12
8. cWorks	13
9. cJobs	14
10. cTickets	15
11. cCustomers	17
<b>SQL Queries</b>	
1. cFinQuery1 (Universal Quantifier)	18
2. cFinQuery2 (Only)	19
3. cFinQuery3 (None)	20
4. cFinQuery4 (Left Join)	21
5. cFinQuery5 (Right Join)	22
6. cFinQuery6 (Full Join)	23
7. cFinQuery7 (6 Tables)	24
8. cFinQuery8 (Moderate Difficulty)	25
9. cFinQuery4 (Moderate Difficulty)	26
10. cFinQuery4 (Moderate Difficulty)	27
<b>References</b>	
1. References	28

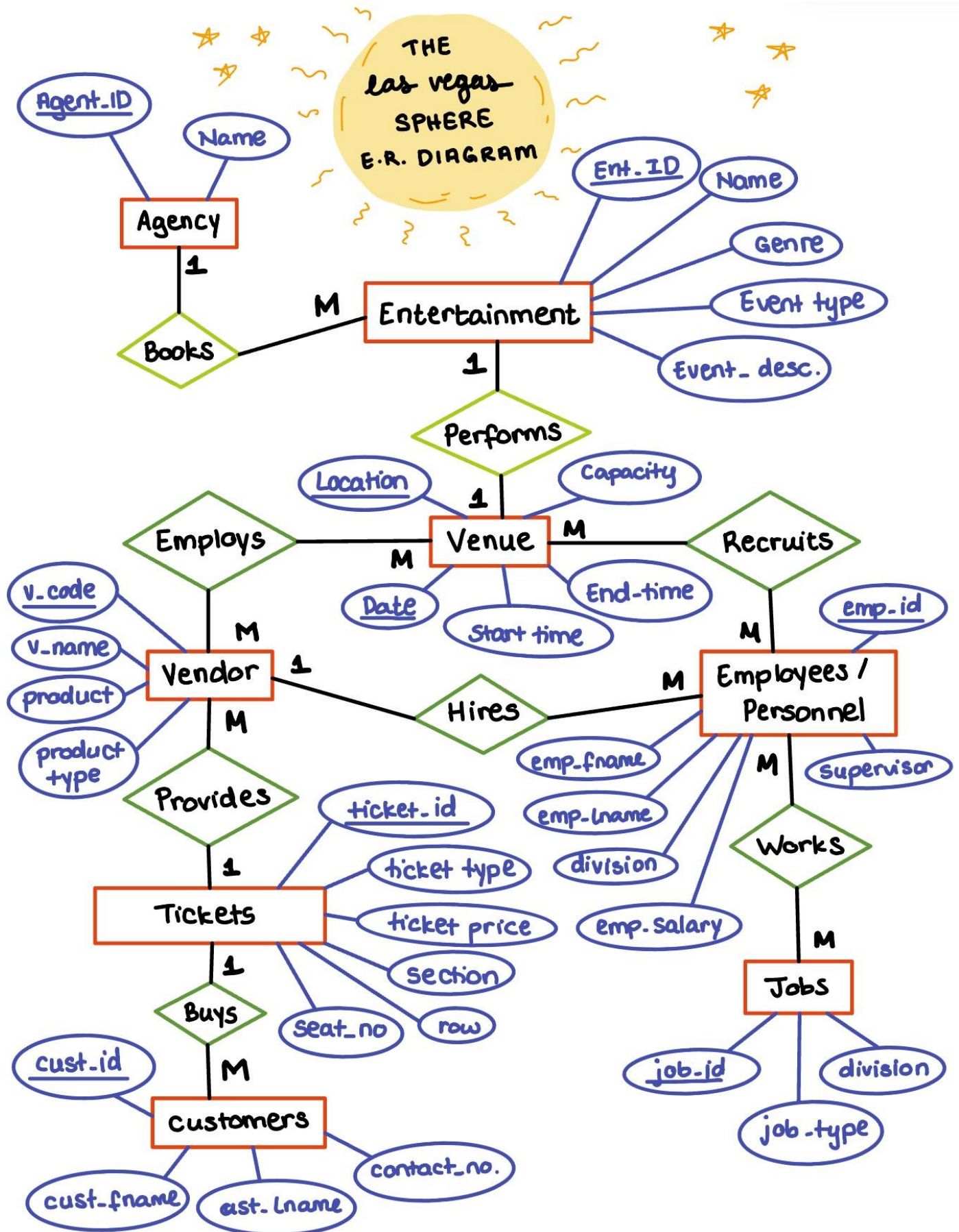
# The Las Vegas Sphere Database

The Sphere is a technological phenomenon located in Las Vegas with the ability to showcase a variety of performances in an immersive view. This 10-year in the making project finally premiered with its debut grand opening in September 2023 and quickly rose to national acclaim due to its verbose technological capacity, light, and audio-visual elements. Consisting of different divisions, seat capacity and multiple levels for venues, this architectural marvel is an entertainment hub. Additionally, it has already gained popularity due to its variety of performances now open to the public as well as influential figures. These performances can range from concerts, immersive experiences, sports showcases, and more and it requires an extensive array of collaboration across different divisions, collaborators and disciplines as defined in the following business rules.

1. Entertainment may only be booked by an agency corresponding in association with them.  
An agency is identified by their ID and also has their name, so they are distinguished.
2. Entertainment, which is identified by their `entertainment_id`, can pertain to different types, have a name, fall into a genre and have a general description of what it provides.
3. Entertainment must be performed only at a certain venue, at a time, depending on its intended capacity, from its beginning to end and on a given date. The venue is identified by the location in the sphere where it will take place.
4. A venue will employ several vendors that give different products. These vendors are identified by their vendor code and also have a vendor name, product and product type.

The vendors can be employed across several venues and these venues can employ multiple vendors.

5. In addition, vendors can provide multiple tickets; however, these tickets come from only one vendor. Tickets are identified by their ticket\_id and also are associated with a ticket type, ticket price, section, row, and seat number for each ticket.
6. Tickets sell a ticket for only one customer, however customers can buy many tickets. Customers are identified by their customer\_id and also are associated with their name and contact number/ information.
7. Venues can also recruit employees/personnel. These employees can work at multiple venues and venues can recruit multiple employees. Employees are identified by their employee\_id and also are associated with their name, division, supervisor, and salary.
8. Employees also work different jobs, and these jobs are worked on by several employees. The jobs are identified by their job\_id and also associated with their job type and the division they are a part of.



## cAgency Table

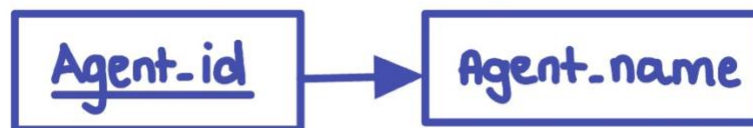
---

This table is about the agencies who will book entertainment for the Sphere.

```
CREATE TABLE "112MUKUNDWAL"."CAGENCY"  
( "AGENT_ID" NUMBER NOT NULL ENABLE,  
"AGENT_NAME" VARCHAR2(20 BYTE),  
CONSTRAINT "CAGENCY_PK" PRIMARY KEY ("AGENT_ID"))
```

FD Diagram:

**cAgency**



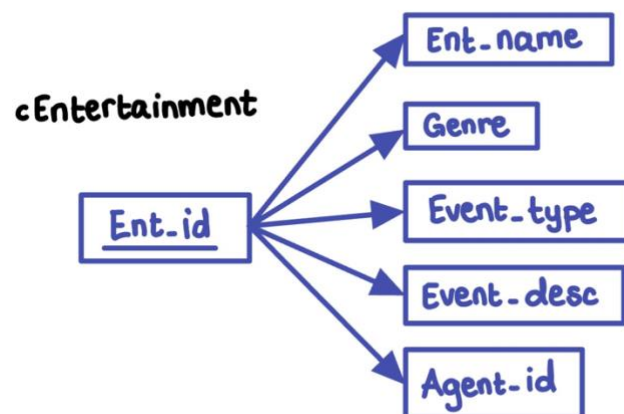
## cEntertainment Table

---

This table is about the entertainment that performs at the Sphere.

```
CREATE TABLE "112MUKUNDWAL"."CENTERTAINMENT"
(
  "ENT_ID" NUMBER,
  "ENT_NAME" VARCHAR2(20 BYTE),
  "GENRE" VARCHAR2(20 BYTE),
  "EVENT_TYPE" VARCHAR2(20 BYTE),
  "EVENT_DESC" VARCHAR2(20 BYTE),
  "AGENT_ID" NUMBER,
  CONSTRAINT "PK_CENTERTAINMENT" PRIMARY KEY ("ENT_ID")
  CONSTRAINT "FK_CENTERTAINMENT_CAGENCY" FOREIGN KEY (
    "AGENT_ID")
    REFERENCES "112MUKUNDWAL"."CAGENCY" ("AGENT_ID") ENABLE
)
```

FD Diagram:



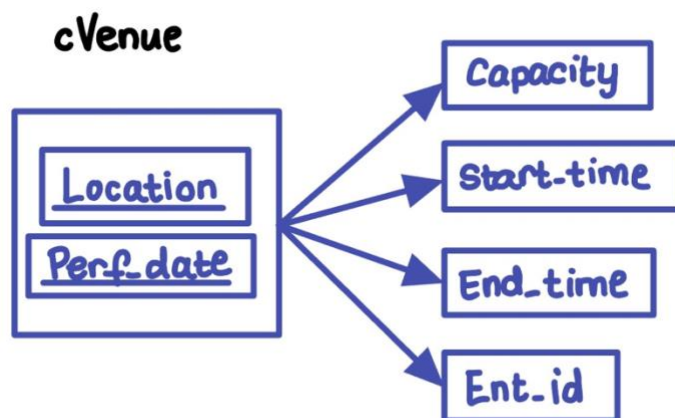
## cVenue Table

---

This table is about the venues where performances can happen, employees can work or vendors sell for.

```
CREATE TABLE "112MUKUNDWAL"."CVENUE"
(
  "LOCATION" VARCHAR2(20 BYTE) NOT NULL ENABLE,
  "PERF_DATE" DATE NOT NULL ENABLE,
  "CAPACITY" NUMBER,
  "START_TIME" VARCHAR2(20 BYTE),
  "END_TIME" VARCHAR2(20 BYTE),
  "ENT_ID" NUMBER,
  CONSTRAINT "CK1_CVENUE" PRIMARY KEY ("LOCATION", "PERF_DATE")
  CONSTRAINT "FK_CVENUE_CENTERTAINMENT" FOREIGN KEY ("ENT_ID")
    REFERENCES "112MUKUNDWAL"."CENTERTAINMENT" ("ENT_ID") ENABLE
)
```

FD Diagram:





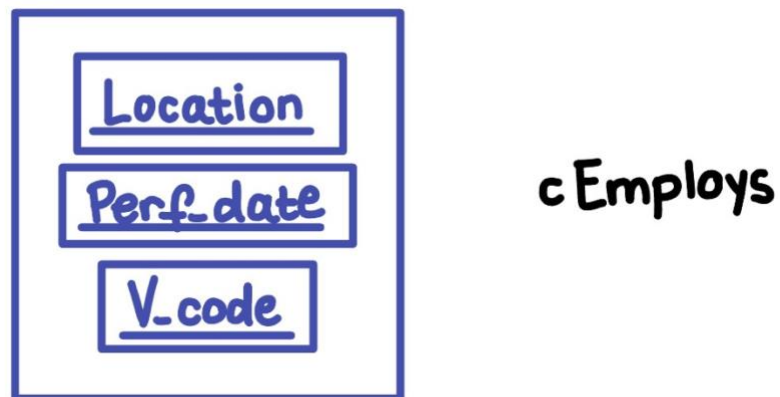
## cEmploys Table

---

This table is about the vendors employed to work at the different venues.

```
CREATE TABLE "112MUKUNDWAL"."CEMPLOYS"  
( "LOCATION" VARCHAR2(20 BYTE) NOT NULL ENABLE,  
  "PERF_DATE" DATE NOT NULL ENABLE,  
  "V_CODE" NUMBER NOT NULL ENABLE,  
  CONSTRAINT "CEMPLOYS_PK" PRIMARY KEY ("LOCATION", "PERF_DATE",  
    "V_CODE")  
  CONSTRAINT "FK_CEMPLOYS_VENDOR" FOREIGN KEY ("V_CODE")  
    REFERENCES "112MUKUNDWAL"."CVENDOR" ("V_CODE") ENABLE  
)
```

FD Diagram:



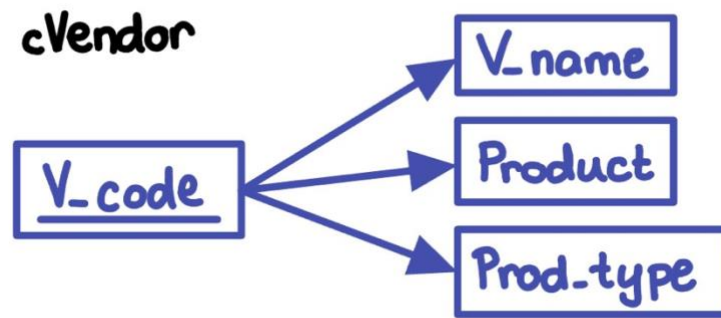
## cVendor Table

---

This table is about the information for the different vendors.

```
CREATE TABLE "112MUKUNDWAL"."CVENDOR"  
( "V_CODE" NUMBER NOT NULL ENABLE,  
  "V_NAME" VARCHAR2(20 BYTE),  
  "PRODUCT" VARCHAR2(20 BYTE),  
  "PROD_TYPE" VARCHAR2(20 BYTE),  
  CONSTRAINT "CVENDOR_PK" PRIMARY KEY ("V_CODE")
```

FD Diagram:



## cRecruits Table

---

This table is about the employees recruited to work at the different venues.

```
CREATE TABLE "112MUKUNDWAL"."CRECRUITS"  
  
  ( "LOCATION" VARCHAR2(20 BYTE) NOT NULL ENABLE,  
    "PERF_DATE" DATE NOT NULL ENABLE,  
    "EMP_ID" NUMBER NOT NULL ENABLE,  
  
    CONSTRAINT "CRECRUITS_PK" PRIMARY KEY ("LOCATION", "PERF_DATE",  
    "EMP_ID")  
  
    CONSTRAINT "FK_CRECRUITS_CVENUE" FOREIGN KEY ("LOCATION",  
    "PERF_DATE")  
  
      REFERENCES "112MUKUNDWAL"."CVENUE" ("LOCATION", "PERF_DATE") ENABLE,  
  
    CONSTRAINT "FK_CRECRUITS_CEMPLOYEES" FOREIGN KEY ("EMP_ID")  
  
      REFERENCES "112MUKUNDWAL"."CEMPLOYEES" ("EMP_ID") ENABLE  
  
  )
```

FD Diagram:



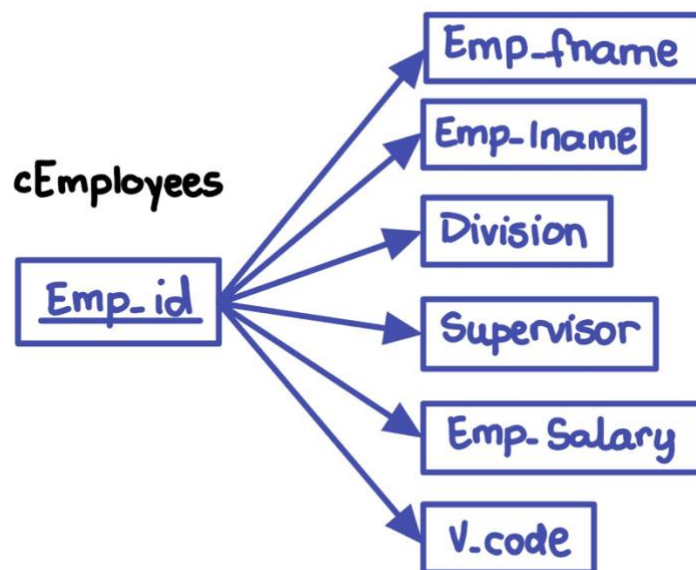
## cEmployees

---

This table is about the information about employees.

```
CREATE TABLE "112MUKUNDWAL"."CEMPLOYEES"  
( "EMP_ID" NUMBER NOT NULL ENABLE,  
  "EMP_FNAME" VARCHAR2(20 BYTE),  
  "EMP_LNAME" VARCHAR2(20 BYTE),  
  "DIVISION" VARCHAR2(20 BYTE),  
  "SUPERVISOR" VARCHAR2(20 BYTE),  
  "EMP_SALARY" VARCHAR2(20 BYTE),  
  "V_CODE" NUMBER,  
  CONSTRAINT "CEMPLOYEES_PK" PRIMARY KEY ("EMP_ID")  ENABLE  
)
```

FD Diagram:



## cWorks Table

---

This table is about the employees that work certain jobs.

```
CREATE TABLE "112MUKUNDWAL"."CWORKS"  
  ( "EMP_ID" NUMBER NOT NULL ENABLE,  
    "JOB_ID" NUMBER NOT NULL ENABLE,  
    CONSTRAINT "CWORKS_PK" PRIMARY KEY ("EMP_ID", "JOB_ID")  
    CONSTRAINT "FK_CWORKS_CEMPLOYEES" FOREIGN KEY ("EMP_ID")  
      REFERENCES "112MUKUNDWAL"."CEMPLOYEES" ("EMP_ID") ENABLE,  
    CONSTRAINT "FK_CWORKS_CJOBS" FOREIGN KEY ("JOB_ID")  
      REFERENCES "112MUKUNDWAL"."CJOBS" ("JOB_ID") ENABLE  
  )
```

FD Diagram:



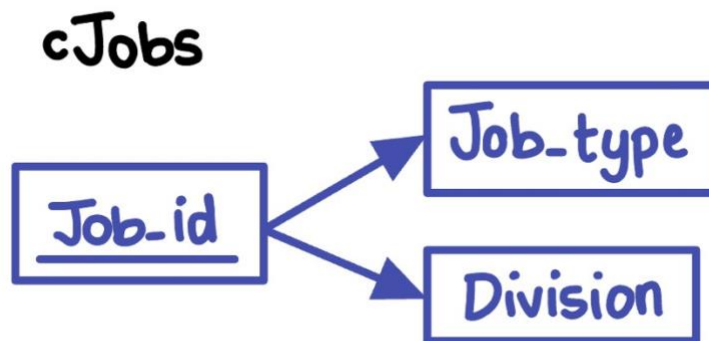
## cJobs Table

---

This is a table about the information available about jobs.

```
CREATE TABLE "112MUKUNDWAL"."CJOBS"  
( "JOB_ID" NUMBER NOT NULL ENABLE,  
  "JOB_TYPE" VARCHAR2(20 BYTE),  
  "DIVISION" VARCHAR2(20 BYTE),  
  CONSTRAINT "CJOBS_PK" PRIMARY KEY ("JOB_ID")  ENABLE  
  )
```

FD Diagram:



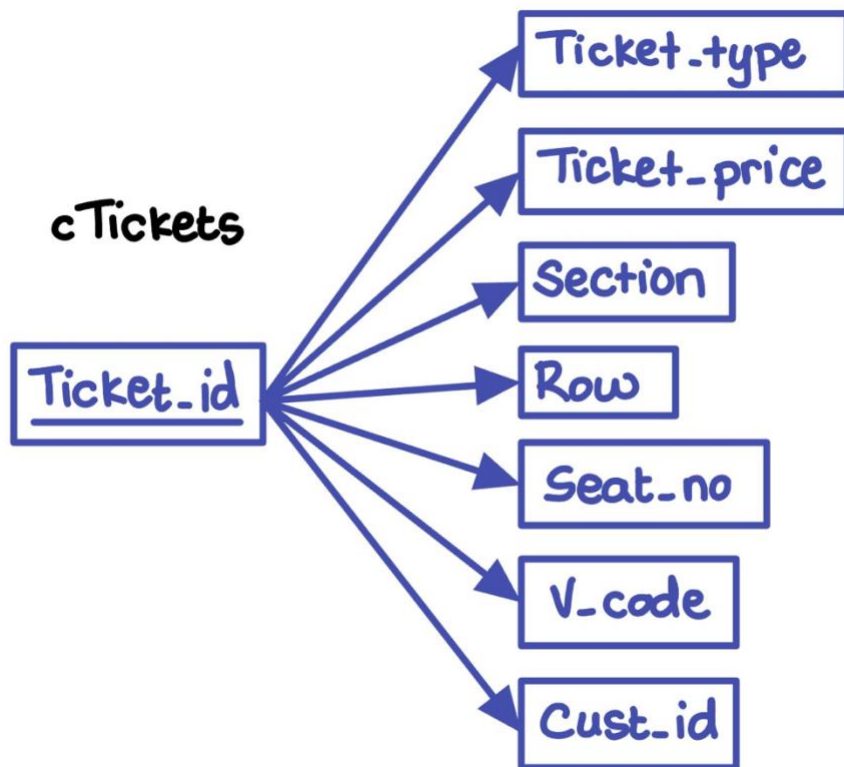
## cTickets Table

---

This is a table about the information about tickets that are bought for entertainment.

```
CREATE TABLE "112MUKUNDWAL"."CTICKETS"  
  (  
    "TICKET_ID" NUMBER NOT NULL ENABLE,  
    "TICKET_TYPE" VARCHAR2(20 BYTE),  
    "TICKET_PRICE" NUMBER,  
    "SECTION" VARCHAR2(20 BYTE),  
    "ROW" NUMBER,  
    "SEAT_NO" NUMBER,  
    "V_CODE" NUMBER,  
    "CUST_ID" NUMBER,  
    CONSTRAINT "CTICKETS_PK" PRIMARY KEY ("TICKET_ID")  
    CONSTRAINT "FK_CTICKETS_CCUSTOMER" FOREIGN KEY ("CUST_ID")  
      REFERENCES "112MUKUNDWAL"."CCUSTOMERS" ("CUST_ID") ENABLE,  
    CONSTRAINT "FK_CTICKETS_CVENDOR" FOREIGN KEY ("V_CODE")  
      REFERENCES "112MUKUNDWAL"."CVENDOR" ("V_CODE") ENABLE  
  )
```

FD Diagram:





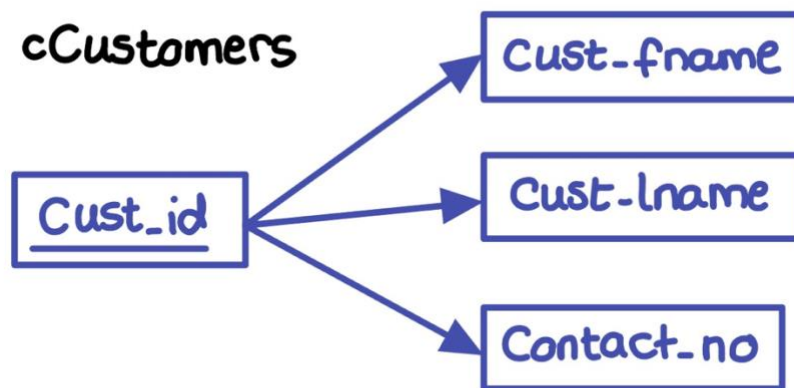
## cCustomers Table

---

This is a table about the information about customers who purchased tickets.

```
CREATE TABLE "112MUKUNDWAL"."CCUSTOMERS"  
( "CUST_ID" NUMBER NOT NULL ENABLE,  
  "CUST_FNAME" VARCHAR2(20 BYTE),  
  "CUST_LNAME" VARCHAR2(20 BYTE),  
  "CONTACT_NO" NUMBER,  
  CONSTRAINT "CCUSTOMERS_PK" PRIMARY KEY ("CUST_ID")  ENABLE  
)
```

FD Diagram:



## cFinQuery1 ~ Universal Quantifier

---

Name customers that purchase a ticket from **every** vendor.

```
SELECT DISTINCT c.cust_fname AS first_name, c.cust_lname AS last_name
FROM cCustomers c
WHERE NOT EXISTS
    (SELECT *
     FROM cVendor v
     WHERE NOT EXISTS
        (SELECT *
         FROM cTickets t
         WHERE t.v_code = v.v_code
         AND t.cust_id = c.cust_id ));
```

	⚡ FIRST_NAME	⚡ LAST_NAME
1	Chase	Soki

Cardinality : 1

## cFinQuery2 ~ Only

---

English: Name the employees who only work only on the Main floor.

```
SELECT DISTINCT emp.emp_fname AS first_name, emp.emp_lname AS
last_name

FROM cEmployees emp

WHERE Emp.emp_Id NOT IN

        (SELECT r.emp_id

        FROM cRecruits r

        WHERE (R.location, R.perf_date) NOT IN

                (SELECT ven.location, ven.perf_date

                FROM cVenue ven

                WHERE Ven.location = 'Main' ));
```

	FIRST_NAME	LAST_NAME
1	Paige	Oliver
2	Avery	Roth
3	Nylah	Berry
4	Ramon	Gamble
5	Flavor	Trey
6	Leilani	Fatim
7	Yurem	Daniels
8	Jaydan	Flores

Cardinality: 8

## cFinQuery3 ~ None

---

English: Name employees who work none of the jobs.

```
SELECT emp.emp_fname AS first_name, emp.emp_lname AS last_name
FROM cEmployees emp
WHERE NOT EXISTS
    ( SELECT j.division
      FROM cJobs j
      WHERE EXISTS
        (SELECT *
         FROM cWorks w
         WHERE w.emp_id = emp.emp_id
               AND w.job_id = j.job_id ));
```

	⚡ FIRST_NAME	⚡ LAST_NAME
1	Paige	Oliver
2	Leilani	Fatim
3	Natalya	Wise
4	Flavor	Trey
5	Avery	Roth

Cardinality: 5

## cFinQuery4 ~ Left Join

---

English: Provide the names of all the agents and the entertainment they book, if any.

```
SELECT DISTINCT a.agent_name, e.ent_name  
FROM cAgency a LEFT JOIN cEntertainment e ON e.agent_id =  
a.agent_id;
```

	AGENT_NAME	ENT_NAME
1	LOL	Drake
2	LOL	Adele
3	LOL	Burna Boy
4	ABC	Formula 1
5	MTV	Interstellar
6	MTV	Oscars
7	MTV	Golden Globes
8	LOL	Kendrick Lamar
9	ABC	World Cup
10	MTV	America's Got Talent
11	LOL	Coldplay
12	NBA	NBA All Stars
13	ABC	Olympics
14	META	Sphere Experience
15	META	Leontyne Price
16	LOL	SZA
17	VVT	(null)
18	LOL	Beyoncé
19	LOL	Travis Scott

Cardinality : 19

## cFinQuery5 ~ Right Join

---

English: Provide the names of employees with their job type and division they work in, if any.

```
SELECT DISTINCT e.emp_fname, e.emp_lname, j.job_type, j.division
FROM cJobs j JOIN cWorks w ON w.job_id = w.job_id RIGHT JOIN
cEmployees e ON e.emp_id = w.emp_id;
```

	EMP_FNAME	EMP_LNAME	JOB_TYPE	DIVISION
34	Omari	Ingram	Immersion Operator	Lights
35	Ramon	Gamble	Lights Engineer	Lights
36	Ramon	Gamble	Sound Technician	Sound
37	Yurem	Daniels	Food Manager	Service
38	Yurem	Daniels	Accounts Overview	Service
39	Yurem	Daniels	Immersion Operator	Lights
40	Halle	Stafford	Lights Engineer	Lights
41	Halle	Stafford	Accounts Overview	Service
42	Halle	Stafford	Immersion Operator	Lights
43	Isabella	Hull	Sercretary	Service
44	Isabella	Hull	Food Manager	Service
45	Isabella	Hull	Accounts Overview	Service
46	Jaydan	Flores	Sercretary	Service
47	Avery	Roth	(null)	(null)
48	Nylah	Berry	Immersion Operator	Lights
49	Charles	Jimenez	Lights Engineer	Lights
50	Charles	Jimenez	Sercretary	Service
51	Charles	Jimenez	Surround Engineer	Sound
52	Leilani	Fatim	(null)	(null)
53	Omari	Ingram	Glow Control	Lights
54	Omari	Ingram	Sound Technician	Sound

Cardinality : 69 only showing 20

## cFinQuery6 ~ Full Join

---

English: Provide the vendor names listed as 'Ticketing' and tickets they sell, if any.

```
SELECT v.v_name, t.ticket_id
FROM cVendor v FULL JOIN cTickets t ON v.v_code = t.v_code
AND v.product = 'Ticketing';
```

	V_NAME	TICKET_ID
1	Ticketmaster	1121341
2	Seat Geek	6759332
3	(null)	6543241
4	Seat Geek	1234512
5	(null)	5467645
6	Ticketmaster	7654256
7	Vivid Seats	8546747
8	Seat Geek	7575689
9	Ticketmaster	5856449
10	Vivid Seats	8984471
11	Seat Geek	4686575
12	Ticketmaster	9674549
13	(null)	6246732
14	Seat Geek	1965925
15	Seat Geek	9503914
16	(null)	1940524
17	Ticketmaster	1446321
18	Ticketmaster	1456432
19	(null)	1457884
20	Networker	(null)
21	Satellite	(null)
22	Stuff Inc	(null)

Cardinality: 22

## cFinQuery7 ~ 6 Tables

---

English: Name the customers who have tickets to see Beyoncé.

```
SELECT c.cust_fname AS first_name, c.cust_lname AS last_name
FROM cCustomers c, cTickets t, cVendor v, cEmploys emp, cVenue ven,
cEntertainment ent
WHERE c.cust_id = t.cust_id
AND t.v_code = v.v_code
AND v.v_code = emp.v_code
AND emp.location = ven.location
AND emp.perf_date = ven.perf_date
AND ven.ent_id = ent.ent_id
AND ent.ent_name = 'Beyoncé';
```

	⚡ FIRST_NAME	⚡ LAST_NAME
1	Felicity	Dudley
2	Camryn	Burnett
3	Kingston	Yoder
4	Chase	Soki
5	Daniel	Webb
6	Kawi	Mays

Cardinality: 6



## cFinQuery8 ~ Moderate Difficulty

---

English: Get ticket prices, the section, entertainment performing and the location and date of the venue for the performance being supplied by Ticketmaster.

```
SELECT DISTINCT t.ticket_price, t.section, e.ent_name, ven.location,  
ven.perf_date
```

```
FROM cTickets t, cVendor v, cCustomers c, cEntertainment e, cEmploys  
em, cVenue ven
```

```
WHERE t.v_code = v.v_code
```

```
AND t.cust_id = c.cust_id
```

```
AND e.ent_id = ven.ent_id
```

```
AND (ven.location , ven.perf_date ) = (( em.location , em.perf_date ))
```

```
AND v.v_code = em.v_code
```

```
AND v.v_name = 'Ticketmaster' ;
```

	TICKET_PRICE	SECTION	ENT_NAME	LOCATION	PERF_DATE
1	122	GA	Beyoncé	Main	03-JUL-23
2	677	VIP	Drake	Main	23-JUN-23
3	676	NBS	Beyoncé	Main	03-JUL-23
4	200	GA	Drake	Main	23-JUN-23
5	677	VIP	Adele	Main	29-JUN-23
6	1378	PIT	Adele	Main	29-JUN-23
7	1378	PIT	Drake	Main	23-JUN-23
8	224	NBS	Drake	Main	23-JUN-23
9	224	NBS	Adele	Main	29-JUN-23
10	122	GA	Adele	Main	29-JUN-23
11	676	NBS	Adele	Main	29-JUN-23
12	200	GA	Adele	Main	29-JUN-23
13	122	GA	Drake	Main	23-JUN-23
14	200	GA	Beyoncé	Main	03-JUL-23
15	677	VIP	Beyoncé	Main	03-JUL-23
16	1378	PIT	Beyoncé	Main	03-JUL-23
17	224	NBS	Beyoncé	Main	03-JUL-23
18	676	NBS	Drake	Main	23-JUN-23

Cardinality : 18

## cFinQuery9 ~ Moderate Difficulty

---

English: Get the location, performance date and capacity for events with employees that work in the light division and entertainment coming from 'LOL' or the ticket vendor is Vivid Seats.

```
SELECT v.location, v.perf_date, v.capacity
FROM cEmployees e, cRecruits r, cVenue v
WHERE e.emp_id = r.emp_id
AND (v.location , v.perf_date ) = (( r.location , r.perf_date ))
AND e.division = 'Lights'
INTERSECT
SELECT v.location, v.perf_date, v.capacity
FROM cAgency a, cVenue v, cEntertainment e
WHERE a.agent_id = e.agent_id
AND e.ent_id = v.ent_id
AND a.agent_name = 'LOL'
UNION
SELECT v.location, v.perf_date, v.capacity
FROM cVendor vend, cEmploys e, cVenue v
WHERE vend.v_code = e.v_code
AND (v.location , v.perf_date ) = (( e.location , e.perf_date ))
AND vend.v_name = 'Vivid Seats' ;
```

	LOCATION	PERF_DATE	CAPACITY
1	Main	29-JUN-23	18000
2	Main	05-JUL-23	18000
3	Main	23-JUL-23	18000
4	Main	27-JUL-23	18000

Cardinality: 4

## cFinQuery10 ~ Moderate Difficulty

---

English: Provide the names of employees who work on the VIP level in the month of June.

```
SELECT DISTINCT emp.emp_fname AS first_name, emp.emp_lname AS  
last_name  
  
FROM cEmployees emp, cRecruits r, cVenue ven  
  
WHERE emp.emp_id = r.emp_id  
  
AND r.location = ven.location  
  
AND r.perf_date = ven.perf_date  
  
AND Ven.location = 'VIP';  
  
AND ven.perf_date BETWEEN '01-Jun-23' AND '01-Jul-23' ;
```

	⚡ FIRST_NAME	⚡ LAST_NAME
1	Charles	Jimenez
2	Halle	Stafford

Cardinality: 2

## References

---

Official Sphere Website: <https://www.thespherevegas.com/>

Cover Photo: <https://www.youtube.com/watch?app=desktop&v=0YVJ4IOGKx4>

Sphere Performance Bookings:

<https://www.venetianlasvegas.com/entertainment/sphere.html#:~:text=Sphere%20opened%20on%20September%2029th,sensory%20film%20Postcard%20from%20Earth.>