**Data Model ERR diagram**

Diagram

Description automatically generated

**Deliverables**

* **Description of my database**

I have used 4 tables to map all requirements to design.

My database is a relational database management system which contains candidate details applying for interview positions and receiving offers. The job positions are advertised by hospitals and interviews are scheduled for the candidates based on their skills and interview role required skillsets.

* **Below are the details of each table and their assumptions**

**hospital**: contains details of all hospitals, address, phone number etc.

1. It is assumed that only 1 hospital of a particular brand is present in one location.

For example: Only 1 “Apollo” hospital is only present in Donegal. Another “Apollo” hospital may be present in Cork.

1. It is assumed all hospital names are different. And recognized by the Hospital\_id attribute

If there are same hospital names, then we can identify them through hospital\_id, telephone number and address

**candidate**: contains details of all candidates, first name, last name, address, phone number, skills etc.

1. It is assumed that - Two candidates may have same name and address. Hence, they can be distinguished by combination of candidate\_id and telephone number
2. It is assumed (`candidate\_id`, `skills`, `telephone`) is used as candidate key since each candidate can have multiple skills
3. For a candidate\_id, only 1 phone number can be tagged.

For example – Two candidates by same name “Nilesh Nayak” and address “Donegal” may have same skills “Python”. However, They can be distinguished as 2 separate candidates because of candidate\_id and telephone numbers being different.

**interview**: contains interview details such as interview id, hospital advertising the interview, candidate attending interview, terview date etc.

1. It is assumed that a candidate can attend

* multiple interviews on the same day advertised by different hospitals
* multiple interviews on the same day advertised by same hospital
* multiple interviews on different days advertised by different hospitals
* multiple interviews on different days advertised by same hospital

1. It is assumed that a candidate can require multiple skills to attend the interview based on job posting
2. Only those candidates registered in the “candidate” table can be called for interviews otherwise relationship will pop a foreign key constraint violation error.
3. It is assumed that a candidate can attend an interview despite him not having matching skills for the position as per requirement
4. Not all candidates having matching skills may be offered a position post attending the interview
5. Only those hospitals registered in the “hospital” table can conduct interviews in their hospitals otherwise relationship will pop a foreign key constraint violation error.
6. “interview\_role” is a foreign key referencing “position\_type” attribute in “position” table

**position:** contains details of all position/role offered, hospital advertising the position, skills required for the position and no. of positions which is up for grabs.

1. Only those hospitals registered in the “hospital” table can advertise open positions in the “position” table.
2. Since a position may have multiple skill requirement, “no\_of\_position” attribute in “position” table may need to be updated across all rows once a position is opened/closed or offer being rolled out.

* **Additions made**

-Initially I used a skill table to match a candidate to skill. However, it was scrapped later to avoid redundancy in data.

-a combination of multiple attributes are chosen to identify candidate keys in “interview” and “position” table to avoid data duplication

-added “offered” attribute in interview table based on requirement

-added interview\_date to capture interview dates based on requirement

-rebuild the index after addition of foreign key constraints

* **Reaction policies used**

I added on update = “cascade” and on delete = “cascade” effect on foreign keys of “interview” and “position” table.

Reason – to maintain data consistency and not violate foreign key constraints with “candidate” and “hospital” table

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

-shared above in page 1

* **Operating system**

windows

* **Constraints added including additional attributes necessary to link the tables**

**Interview Table**

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

**Position Table**

Graphical user interface, application

Description automatically generated

**Codes for adding above constraints**

ALTER TABLE `nayak21200475`.`position`

ADD CONSTRAINT `hosp\_id`

FOREIGN KEY (`hosp\_id\_advertising\_position`)

REFERENCES `nayak21200475`.`hospital` (`hospital\_id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION;

ALTER TABLE `nayak21200475`.`interview`

ADD CONSTRAINT `interview\_candidate`

FOREIGN KEY (`interview\_candidate\_id`)

REFERENCES `nayak21200475`.`candidate` (`candidate\_id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION;

ALTER TABLE `nayak21200475`.`interview`

ADD CONSTRAINT `interview\_hospital`

FOREIGN KEY (`interview\_hospital\_id`)

REFERENCES `nayak21200475`.`hospital` (`hospital\_id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION;

ALTER TABLE `nayak21200475`.`interview`

ADD CONSTRAINT `interview\_position`

FOREIGN KEY (`interview\_role`)

REFERENCES `nayak21200475`.`position` (`position\_type`)

ON DELETE NO ACTION

ON UPDATE NO ACTION;

* **Appropriate data types for all attributes and primary key(s).**

Highlighted in yellow are primary & candidate keys applied to each of the table

CREATE TABLE `**nayak21200475`.`hospital**` (

`hospital\_id` VARCHAR(45) NOT NULL,

`hospital\_name` VARCHAR(45) NOT NULL,

`telephone` INT NOT NULL,

`address` VARCHAR(45) NOT NULL,

PRIMARY KEY (`hospital\_id`));

CREATE TABLE `**nayak21200475`.`candidate**` (

`candidate\_id` VARCHAR(45) NOT NULL,

`firstname` VARCHAR(45) NOT NULL,

`surname` VARCHAR(45) NOT NULL,

`address` VARCHAR(45) NOT NULL,

`telephone` INT NOT NULL,

`skills` VARCHAR(45) NOT NULL,

PRIMARY KEY (`candidate\_id`, `skills`, `telephone`));

CREATE TABLE `**nayak21200475`.`position`** (

`position\_id` VARCHAR(45) NOT NULL,

`position\_type` VARCHAR(45) NOT NULL,

`hosp\_id\_advertising\_position` VARCHAR(45) NOT NULL,

`skills\_required` VARCHAR(45) NOT NULL,

`no\_of\_positions` INT NOT NULL,

PRIMARY KEY (`position\_id`, `skills\_required`, `position\_type`, `hosp\_id\_advertising\_position`));

CREATE TABLE `**nayak21200475`.`interview**` (

`interview\_id` VARCHAR(45) NOT NULL,

`interview\_role` VARCHAR(45) NOT NULL,

`interview\_date` DATE NOT NULL,

`interview\_candidate\_id` VARCHAR(45) NOT NULL,

`interview\_hospital\_id` VARCHAR(45) NOT NULL,

`offered` VARCHAR(45) NOT NULL,

PRIMARY KEY (`interview\_id`));

* **Stored proc for populating tables**

Text

Description automatically generated

**Codes below**

CREATE PROCEDURE `enter\_position\_table\_details`(IN position\_id VARCHAR(45), IN position\_type VARCHAR(45), IN hosp\_id\_advertising\_position VARCHAR(45), IN skills\_required VARCHAR(45), IN no\_of\_positions INT)

BEGIN

INSERT INTO `nayak21200475`.`position` (`position\_id`, `position\_type`, `hosp\_id\_advertising\_position`, `skills\_required`, `no\_of\_positions`)

VALUES ( position\_id, position\_type, hosp\_id\_advertising\_position, skills\_required, no\_of\_positions);

END

CREATE PROCEDURE `enter\_interview\_table\_details` (IN interview\_id VARCHAR(45), IN interview\_role VARCHAR(45), IN interview\_date DATE, IN interview\_candidate\_id VARCHAR(45),

IN interview\_hospital\_id VARCHAR(45), IN offered VARCHAR(45))

BEGIN

INSERT INTO `nayak21200475`.`interview` (`interview\_id`,`interview\_role`,`interview\_date`,`interview\_candidate\_id`,`interview\_hospital\_id`,`offered`)

VALUES (interview\_id,interview\_role,interview\_date,interview\_candidate\_id,interview\_hospital\_id,offered);

END

CREATE PROCEDURE `enter\_candidate\_table\_details`(IN candidate\_id VARCHAR(45), IN firstname VARCHAR(45), IN surname DATE, IN address VARCHAR(45), IN telephone INT, IN skills VARCHAR(45))

BEGIN

INSERT INTO `nayak21200475`.`candidate` (`candidate\_id`,`firstname`,`surname`,`address`,`telephone`,`skills`)

VALUES (candidate\_id , firstname , surname, address, telephone, skills);

END

CREATE PROCEDURE `enter\_hospital\_table\_details`(IN hospital\_id VARCHAR(45), IN hospital\_name VARCHAR(45), IN address VARCHAR(45), IN telephone INT)

BEGIN

INSERT INTO `nayak21200475`.`hospital` (`hospital\_id`, `hospital\_name`, `address`, `telephone`)

VALUES (hospital\_id, hospital\_name, address, telephone);

END

* **Stored proc without parameterized queries (5,7,8,11)**

call nayak21200475.query5();

call nayak21200475.query7();

call nayak21200475.query8();

call nayak21200475.query11();

Note: query<number> relates to questions asked in Step 4

* **Stored proc with parameterized queries (1,2,3,4,6,9,10)**

call nayak21200475.query1('H001');

call nayak21200475.query2('Apollo');

call nayak21200475.query3('Nayak');

call nayak21200475.query4('10004');

call nayak21200475.query6('communication');

call nayak21200475.query9('2021-01-01');

call nayak21200475.query10('2021-01-01');

Note: query<number> relates to questions asked in Step 4

**Insight:** In parameterized stored procedure, input parameter name should not be same as column name in the database.

It will pop out all rows in underlying table.

**Codes for each of the 11 stored procedure (some having parametric queries) in Step 4**

CREATE PROCEDURE `query1` (IN hosp\_id varchar(45))

BEGIN

select hospital\_name from `nayak21200475`.`hospital` where hospital\_id = hosp\_id;

END

CREATE PROCEDURE `query2` (IN hosp\_name varchar(45))

BEGIN

select \* from `nayak21200475`.`hospital` where hospital\_name = hosp\_name;

END

CREATE PROCEDURE `query3` (in sur\_name varchar(45))

BEGIN

select \* from `nayak21200475`.`candidate` where surname = sur\_name;

END

CREATE PROCEDURE `query4`(in pos\_id varchar(45))

BEGIN

select distinct candidate\_id, firstname, surname from candidate where skills in (select skills\_required from `nayak21200475`.`position` where position\_id = pos\_id);

END

CREATE PROCEDURE `query5`()

BEGIN

select count(distinct interview\_candidate\_id) from interview where offered = 'yes';

END

CREATE PROCEDURE `query6` (in skills\_reqd varchar(45))

BEGIN

select position\_type from position where skills\_required = skills\_reqd;

END

CREATE PROCEDURE `query7`()

BEGIN

select sum(b.no\_of\_positions) as no\_of\_positions from (select distinct position\_id, no\_of\_positions from position where position\_type = "nurse") b;

END

CREATE PROCEDURE `query8`()

BEGIN

select h.hospital\_name, p.position\_type from position p , hospital h where p.hosp\_id\_advertising\_position = h.hospital\_id group by 1 order by 1 asc, 2 asc;

END

CREATE PROCEDURE `query9` (in intview\_dt date)

BEGIN

select \* from interview where `nayak21200475`.interview\_date = intview\_dt;

END

CREATE PROCEDURE `query10` (in intview\_dt date)

BEGIN

select distinct interview\_candidate\_id from interview where `nayak21200475`.interview\_date = intview\_dt;

END

CREATE PROCEDURE `query11` ()

BEGIN

select distinct b.interview\_candidate\_id , c.firstname, c.surname

from candidate c join

(select interview\_candidate\_id, count(\*) from interview

group by interview\_candidate\_id

having count(\*) >= 2) b

on b.interview\_candidate\_id = c.candidate\_id;

END

**Few additional tests checked after creating databases and adding constraints**

**Test 1**

**To test primary key and foreign key constraints**, I add new entries in my base tables using stored procedure

It successfully added the new rows.

call nayak21200475.enter\_candidate\_table\_details ('C0012','Miley', 'Jones', 'Wicklow', '9999916', 'active listening');

call nayak21200475.enter\_hospital\_table\_details ('H025','Kidneycare', 'Donegal', '888825');

call nayak21200475.enter\_interview\_table\_details ('I113','Virologist', '2021-01-01', 'C1010', 'H007', 'no');

call nayak21200475.enter\_position\_table\_details ('10011','Virologist', 'H001', 'investigation', '1');

call nayak21200475.enter\_position\_table\_details ('10011','Virologist', 'H001', 'critical thinking', '1');

**Test 2**

**To check duplicate entries** being added in my base tables using stored procedure, I used below queries and tried to see if my primary/candidate keys and table constraints are working as expected.

call nayak21200475.enter\_candidate\_table\_details ('C005','Mikael', 'Souza', 'Dublin', '9999905', 'active listening');

call nayak21200475.enter\_hospital\_table\_details ('H001','Apollo', 'Donegal', '888801');

call nayak21200475.enter\_interview\_table\_details ('I113','Virologist', '2021-01-01', 'C1010', 'H007', 'yes');

call nayak21200475.enter\_position\_table\_details ('10001','Virologist', 'H001', 'investigation', '1');

It successfully popped – “Duplicate entry” error as shown below.

