Reports table SQL:

-- Sequence: music\_school.reports\_seq

-- DROP SEQUENCE music\_school.reports\_seq;

CREATE SEQUENCE music\_school.reports\_seq

INCREMENT 1

MINVALUE 1

START 1

CACHE 20;

ALTER TABLE music\_school.reports\_seq

OWNER TO gzlrnynoieypkd;

-- Table: music\_school.reports

-- DROP TABLE music\_school.reports;

CREATE TABLE music\_school.reports

(

id integer NOT NULL DEFAULT nextval('music\_school.reports\_seq'::regclass),

name character varying(45) NOT NULL,

CONSTRAINT reports\_pkey PRIMARY KEY (id),

CONSTRAINT reports\_name\_uq UNIQUE (name)

)

WITH (

OIDS=FALSE

);

ALTER TABLE music\_school.reports

OWNER TO gzlrnynoieypkd;

Database changes:

Remove these fields from TeacherApplicants:

address, **(Was removed in the last DB creation)**

resumeLink,

passwordId **(Was removed in the last DB creation)**

Add these fields to TeacherApplicants:

coverletter (very long string),

hours (int)

Change these fields on TeacherApplicants:

Status varchar into foreign key of response\_status table. **(Was done in the last DB creation - called ‘status\_id’ - foreign key to request\_status table)**

Add two tables:

1. teacherApplicantReferences:

teacherApplicantId (int),

name (string),

phonenumber (int) - needs to be a unique combination of all 3 fields as can have multiple references for same teacher applicant **(Made all 3 primary keys. I believe that is what you want? Primary keys are automatically unique)**

2. teacherApplicantExperience:

teacherApplicantId (int),

instrument (string),

grade (int) - again,

unique combination of all 3 for the same reason **(Made all 3 primary keys. I believe that is what you want? Primary keys are automatically unique)**

All base ids (eg student\_id should just be id)

insthire\_requests:

request\_id -> id

request\_status -> request\_status\_id (int - foreign key)

approved\_date -> date (not timestamp - not suitable datatype)

**New table for request\_status’ (id(int), status(varchar))**

instrument\_hire

inst\_hire\_id -> id

add is\_returned: bool

student\_experience:

inst\_type\_id -> Int

grade -> int

Remove exp\_description

instrument\_types:

inst\_type\_id -> id: Int

Added: lesson\_fee (decimal)

parents:

parent\_id -> id

surname -> last\_name

email -> Unique

students:

student\_id -> id

surname -> last\_name

email -> Unique

managers:

manager\_id -> id

surname -> last\_name

email -> Unique

passwords:

password\_id -> id

instruments:

instrument\_id -> id

serial\_no -> Unique

inst\_type\_id-> int

condition->condition\_id: int (foreign key)

**Create conditions table (id, condition)**

description->model

rooms:

Remove instruments\_suited

Add name: Varchar

Add is\_available: bool

Remove room\_notes

**NOTE: room\_no is NOT auto increment**

teacher\_experience:

inst\_type\_id -> int

grade -> int

exp\_description -> Nullable

teachers:

teacher\_id->id

surname -> last\_name

email -> Unique

lessons:

lesson\_id -> id

inst\_type\_id -> int

lesson\_requests:

request\_id->Id

inst\_type\_id -> Int

Remove lesson\_id

Add teacher\_id -> id

request\_status -> request\_status\_id (int - same table as earlier)

approved\_date -> date (not timestamp - not suitable datatype)

teacher\_contracts:

contract\_id -> id

employment\_type->employment\_type\_id (int)

Create table employment\_types (id(int), employment\_type(varchar))

salary -> Decimal

Requires validation that either salary or pay\_rate is filled.

teacher\_change\_requests

request\_id -> id

Add lesson\_id -> foreign key

request\_status -> request\_status\_id (Same as before)

teacher\_applicants

applicant\_id -> id

surname -> last\_name

status -> status\_id (same as before - request\_status table)

Remove: middleName, dob, address, passwordId

resumeLink -> Not Nullable

email -> Unique

manager\_contracts (Same as teacher\_contracts)

contract\_id -> id

employment\_type -> employment\_type\_id (int - foreign key)

salary -> Decimal

Requires validation that either salary or pay\_rate is filled.