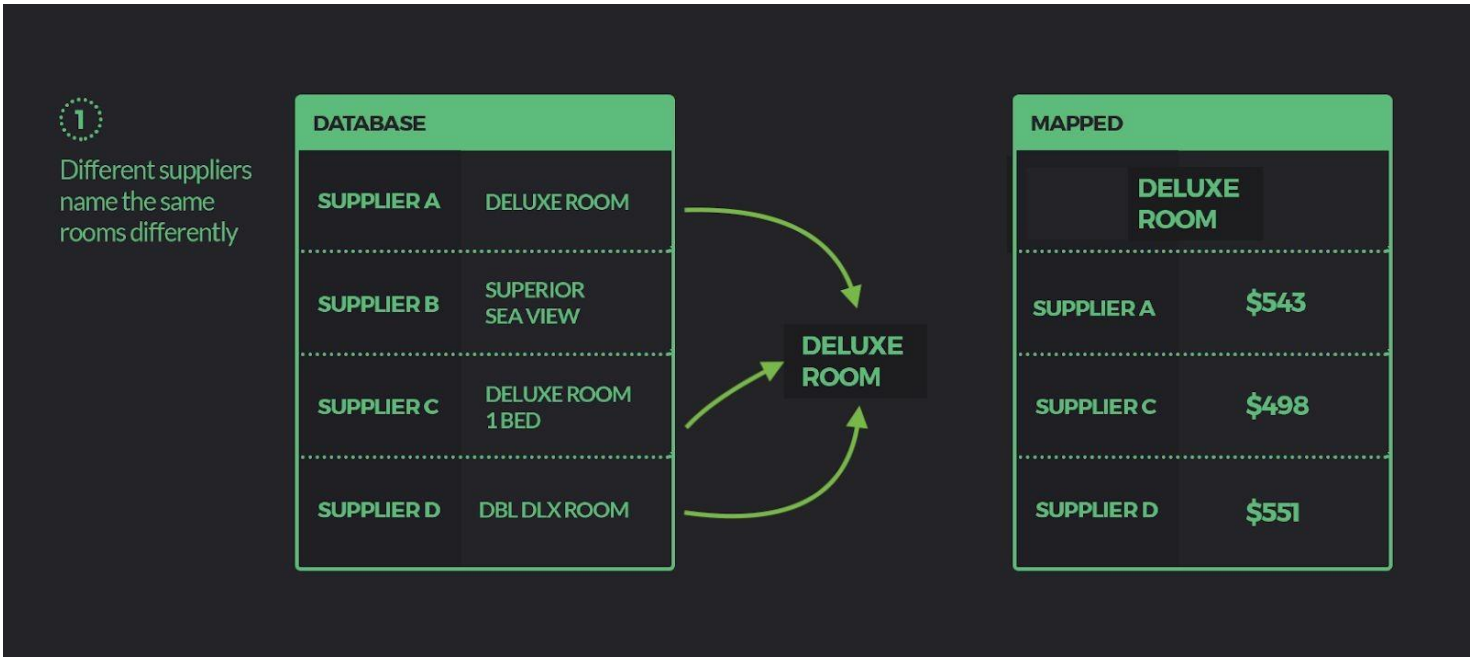
Assignment 1

You are part of the development team that is building a marketplace for searching and booking Hotel Rooms. The marketplace fetches Hotels from various online platforms(like Agoda, Booking.com, Expedia etc.) and shows the best available rates from these hotels to the users to book. The requirement demands that the platform show all the different room types that can be booked for a hotel when a user clicks into a specific hotel and wants to book a room. This means that the Room types shown should be a superset/union of the various rooms fetched from various sources.

A core problem the team is facing is that they display duplicate room types since the Room types might not be defined or labeled the same way across various Online platforms(Suppliers). See Example below, The same room is labelled differently in 3 of the 4 different suppliers. What we need to do is identify that and map them all to one room called DELUXE ROOM in our system and tag the different prices (for the same room type) from different suppliers to this one room type, instead of displaying them as 4 different room types

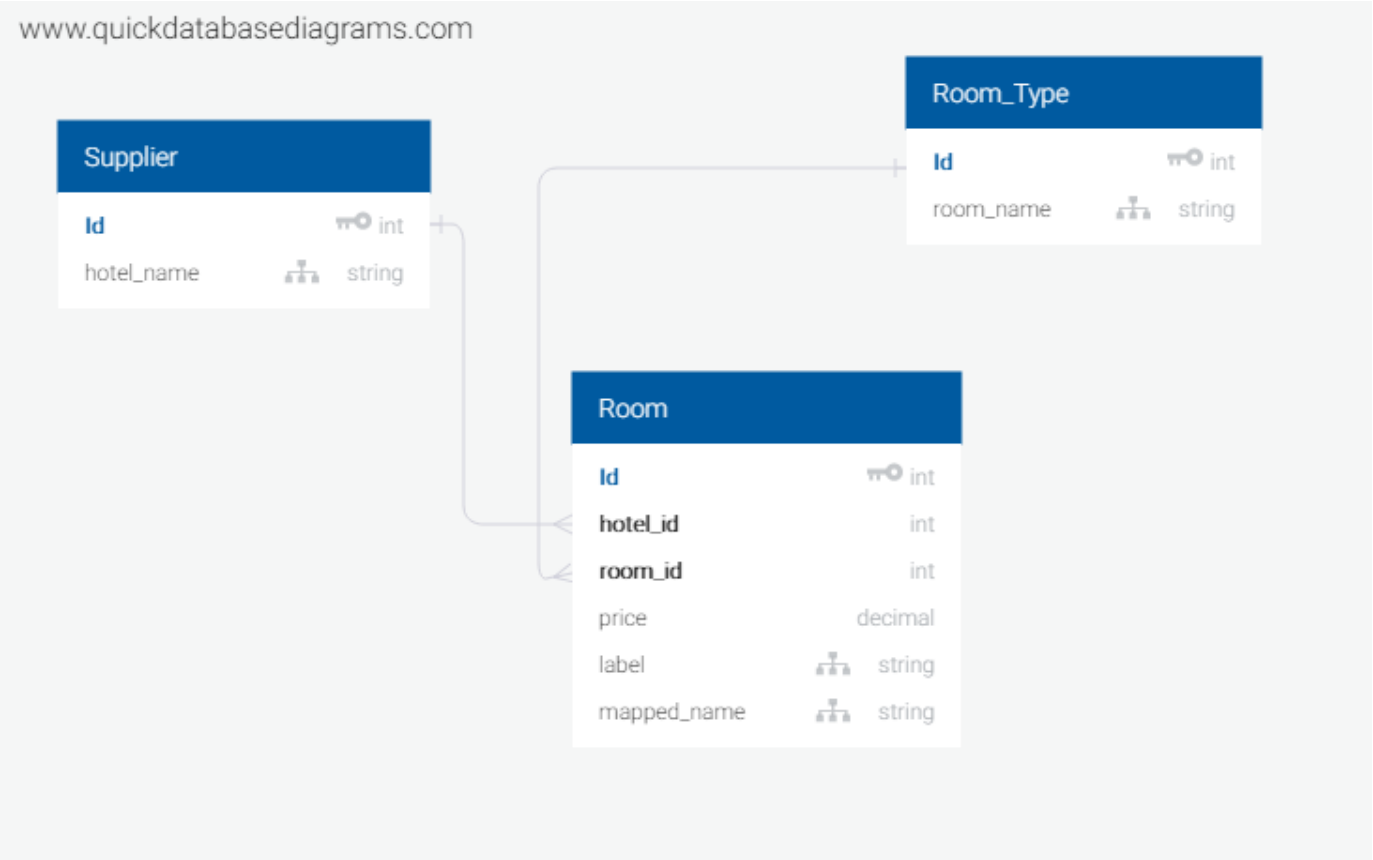


Assignment: So if you were to come up with a creative solution that would identify and remove the duplicates(or rather merge them) of room types how would you go about it?

Please note that we are not looking for an exact solution or a big architecture/design here. All we want to understand is your thought process and how you would approach the problem.

Solution

Database Diagram



Database Table Schema

```
create table Supplier (
 Id int primary key,
 hotel_name varchar(100)
);
create table Room_Type (
 Id int primary key,
 room_name varchar(100)
);
create table Room (
 Id int primary key,
 hotel_id int not null,
 room_id int not null ,
 price decimal not null,
 label varchar(100)
);
```

Sample Records

```
/*Supplier Sample Records */
insert into Supplier values(1, 'Supplier A');
insert into Supplier values(2, 'Supplier B');
insert into Supplier values(3, 'Supplier C');
insert into Supplier values(4, 'Supplier D');

/*Room Type Sample Records*/
insert into Room_Type values(1, 'DELUXE ROOM');
insert into Room_Type  values(2, 'SUPERIOR ROOM');

/*deluxe room sample records */
insert into  Room values(1, 1, 1, 543, 'DELUXE ROOM' );
insert into  Room values(3, 3, 1, 498, 'DELUXE ROOM 1 BED');
insert into  Room values(4, 4, 1, 551, 'DBLDLX ROOM');

/*superior room sample records*/
insert into  Room values(5, 3, 2, 280, 'SUPER ROOM');
insert into  Room values(6, 4, 2, 410, 'SUP-ROOM');
insert into  Room values(2, 2, 2, 543, 'SUPERIOR ROOM');
```

Filter query

```
/* to select deluxe room */
select r.*, s.hotel_name, rt.room_name from Room r
 join Supplier s
 on s.Id = r.hotel_id
   inner join Room_Type rt
on r.room_id = rt.Id
where r.room_id = 1 /* room*/
group by r.hotel_id;
```

Result

Id	hotel_id	room_id	price	label	hotel_name	room_name
1	1	1	543	DELUXE ROOM	Supplier A	DELUXE ROOM
3	3	1	498	DELUXE ROOM 1 BED	Supplier C	DELUXE ROOM
4	4	1	551	DBLDLX ROOM	Supplier D	DELUXE ROOM

```
/* to select superior room */
select r.*, s.hotel_name, rt.room_name from Room r
 join Supplier s
 on s.Id = r.hotel_id
   inner join Room_Type rt
on r.room_id = rt.Id
```

```
where r.room_id = 2
group by r.hotel_id;
```

output

Id	hotel_id	room_id	price	label	hotel_name	room_name
2	2	2	543	SUPERIOR ROOM	Supplier B	SUPERIOR ROOM
5	3	2	280	SUPER ROOM	Supplier C	SUPERIOR ROOM
6	4	2	410	SUP-ROOM	Supplier D	SUPERIOR ROOM

ult 27 ×