1. A. Normalisasi

**UNF (Unnormalized Form):**

1. Tabel "staff": staffid, namee, email, phonenumber, work
2. Tabel "signup": userid, username, email, pass
3. Tabel "emp\_login": loginid, email, password
4. Tabel "room": roomid (PK), roomtype, bedding
5. Tabel "roombook": bookingid, roomid, userid , namee, email, country, phone, roomtype, bed, meal, noofroom, cin, cout, nodays, stat
6. Tabel "payment": id, Namee, Email, bookingid, NoofRoom, cin, cout, noofdays, roomtotal, bedtotal, meal, mealtotal, finaltotal

**PENJELASAN UNF**

Dalam UNF (Unnormalized Form) diatas, terdapat beberapa tabel dengan struktur yang belum dinormalisasi untuk dapat mengurangi duplikasi data dan memastikan struktur basis data yang lebih optimal. .

- Tabel "staff" berisi informasi staf dengan kolom-kolom seperti identifikasi staf, nama, alamat email, nomor telepon, dan pekerjaan.

- Tabel "signup" berisi informasi pengguna yang mendaftar dengan kolom-kolom seperti identifikasi pengguna, nama pengguna, alamat email, dan kata sandi.

- Tabel "emp\_login" berisi entri login pegawai dengan kolom-kolom seperti identifikasi entri login, alamat email, dan kata sandi. Tabel "room" berisi informasi tentang kamar-kamar yang tersedia dengan kolom-kolom seperti identifikasi kamar (kunci primer), jenis kamar, dan jenis tempat tidur.

- Tabel "roombook" berisi informasi pemesanan kamar dengan kolom-kolom seperti identifikasi pemesanan, identifikasi kamar yang dipesan, identifikasi pengguna yang memesan, nama pemesan, alamat email pemesan, negara, nomor telepon, jenis kamar, jenis tempat tidur, makanan, jumlah kamar, tanggal check-in, tanggal check-out, jumlah hari, dan status.

- Tabel "payment" berisi informasi pembayaran untuk pemesanan kamar dengan kolom-kolom seperti identifikasi, nama pembayar, alamat email pembayar, identifikasi pemesanan, jumlah kamar, tanggal check-in, tanggal check-out, jumlah hari, total biaya kamar, total biaya tempat tidur, makanan, total biaya makanan, dan total biaya akhir.

**1NF (First Normal Form)**

1. Tabel "staff":  
- staff(staffid PK, namee, email, phonenumber, work)

2. Tabel "signup":  
- signup(userid PK, username, email, pass)

3. Tabel "emp\_login":  
- emp\_login(loginid PK FK signup.userid, FK staff.staffid, PK ,

email, password)

4. Tabel "room":

- room(roomid PK, roomtype, bedding) 5. Tabel "roombook":

- Roombook (bookingid FK roombook.bookingid, roomid FK room.roomid, userid FK signup.userid, namee, email, country, phone, roomtype, bed, meal, noofroom, cin, cout, nodays, stat)

6. Tabel "payment":  
- payment(id PK, Namee, Email, bookingid FK

roombook.bookingid, NoofRoom, cin, cout, noofdays, roomtotal, bedtotal, meal, mealtotal, finaltotal)

**Penjelasan 1NF**

Setelah melakukan normalisasi ke tingkat pertama (1NF), tabel-tabel dalam basis data diperbarui sebagai berikut:

1. Tabel "staff":
   * Kolom staffid merupakan kunci primer (PK) yang digunakan untuk mengidentifikasi secara unik setiap staf.
2. Tabel "signup":.
   * Kolom userid adalah kunci primer (PK) yang digunakan untuk mengidentifikasi setiap entri pendaftaran secara unik.
3. Tabel "emp\_login":
   * Kolom loginid adalah kunci primer (PK) yang mengidentifikasi secara unik setiap entri login pegawai.
   * Kolom email dan password berfungsi sebagai kunci asing (FK) yang merujuk ke tabel "signup" untuk mengaitkan entri login dengan entri pendaftaran yang sesuai.
4. Tabel "room":
   * Tabel "room" tetap tidak berubah karena strukturnya sudah memenuhi 1NF sebelumnya.
   * Kolom roomid adalah kunci primer (PK) yang mengidentifikasi setiap kamar secara unik.
5. Tabel "roombook":
   * Tabel "roombook" telah dinormalisasi menjadi 1NF.
   * Kolom bookingid adalah kunci primer (PK) yang mengidentifikasi setiap pemesanan kamar secara unik.
   * Kolom roomid, userid mengandung kunci asing (FK) yang merujuk ke tabel "room" dan "signup" masing-masing.
6. Tabel "payment":
   * Kolom id adalah kunci primer (PK) yang mengidentifikasi setiap pembayaran secara unik.
   * Kolom bookingid mengandung kunci asing (FK) yang merujuk ke tabel "roombook" untuk menghubungkan pembayaran dengan pemesanan yang sesuai.

**2NF (Second Normal Form):**

1. Table Tabel “staff":  
   - staff(staffid (PK), namee, email, phonenumber, work)
2. Tabel “signup":  
   - signup(userid (PK), username, email, pass)
3. Tabel "emp\_login":  
   - emp\_login(loginid (PK, FK), email, password)

FOREIGN KEY (loginid) REFERENCES signup(userid)

FOREIGN KEY (loginid) REFERENCES staff(staffid)

1. Tabel "room":

- room(roomid (PK), roomtype, bedding)

1. Tabel "roombook":

* roombook(bookingid (PK), roomid (FK room.roomid), userid (FK signup.userid), namee, email, country, phone, roomtype, bed, meal, noofroom, cin, cout, nodays, stat)

1. Tabel "payment":  
   - payment(id (PK), Namee, Email, bookingid (FK

roombook.bookingid), NoofRoom, cin, cout, noofdays, roomtotal, bedtotal, meal, mealtotal, finaltotal)

**PENJELASAN 2NF**

Setelah normalisasi ke tingkat kedua (2NF), berikut tabel-tabel dalam basis data :

Tabel "staff" tetap sama seperti sebelumnya dengan struktur yang memenuhi 2NF. Kolom staffid masih berfungsi sebagai kunci primer (PK) yang mengidentifikasi setiap entitas staf.

Tabel "signup" juga mempertahankan struktur yang sama dan memenuhi 2NF. Kolom userid berfungsi sebagai kunci primer (PK) yang unik untuk setiap entri pendaftaran.

Tabel "emp\_login" telah diubah dengan penambahan kunci asing (FK) dan referensi. Kolom loginid masih merupakan kunci primer (PK) yang unik untuk setiap entri login pegawai. Kolom email dan password tetap ada, dan kolom loginid merujuk ke kolom userid di tabel "signup" dan juga merujuk ke kolom staffid di tabel “staff".

Tabel "room" tidak mengalami perubahan dalam struktur dan tetap memenuhi 2NF. Kolom roomid masih berfungsi sebagai kunci primer (PK), sedangkan kolom roomtype dan bedding berisi informasi tentang jenis kamar dan jenis tempat tidur.

Tabel "roombook" juga tetap sama seperti sebelumnya dan memenuhi 2NF. Kolom bookingid berfungsi sebagai kunci primer (PK) yang unik untuk setiap pemesanan kamar. Kolom roomid dan userid masih berfungsi sebagai kunci asing (FK) yang merujuk ke tabel "room" dan "signup" masing-masing.

Tabel "payment" tetap memenuhi 2NF setelah normalisasi. Kolom id adalah kunci primer (PK) yang unik untuk setiap pembayaran. Kolom bookingid merupakan kunci asing (FK) yang merujuk ke kolom bookingid di tabel "roombook" untuk menghubungkan pembayaran dengan pemesanan yang sesuai.

**3NF (Third Normal Form)**

1. Tabel "staff":  
   - staff(staffid (PK), namee, email, phonenumber, work)
2. Tabel "signup":

-  signup(userid (PK), email,)

-  userinfo (userid FK signup.userid, username, pass)   
FOREIGN KEY (userid) REFERENCES signup(userid)

1. Tabel "emp\_login":

- emp\_login(loginid (PK, FK), email, password)

FOREIGN KEY (loginid) REFERENCES signup(userid)

FOREIGN KEY (loginid) REFERENCES staff(staffid)

4. Tabel "room":

- room(roomid (PK), roomtype, bedding)

1. Tabel "roombook":

-  roombook(bookingid (PK), roomid (FK), userid (FK), namee, email, country, phone, stat)

- booking\_info(bookingid (FK), roomtype, bed, meal, noofroom, cin, cout, nodays)  
FOREIGN KEY (bookingid) REFERENCES roombook(bookingid)

1. Tabel "payment":

-  payment(id (PK), Namee, Email, bookingid (FK), NoofRoom, cin,   
cout, noofdays, finaltotal)  
FOREIGN KEY (bookingid) REFERENCES roombook(bookingid)

-  roombooking(bookingid (PK), roomtotal, bedtotal, meal, mealtotal)

**Penjelasan 3NF**

Setelah normalisasi ke tingkat ketiga (3NF), berikut tabel-tabel dalam basis data :

Tabel "staff" tetap sama dengan struktur yang memenuhi 3NF. Kolom staffid berfungsi sebagai kunci primer (PK) yang mengidentifikasi setiap entitas staf. Kolom namee, email, phonenumber, dan work berisi informasi terkait staf tersebut.

Tabel "signup" telah diubah dengan penambahan tabel terkait. Kolom userid masih berfungsi sebagai kunci primer (PK), dan kolom email berisi informasi email dari pengguna. Tabel terkait "userinfo" mengandung kolom username dan pass yang berhubungan dengan entri signup melalui kolom userid yang berfungsi sebagai kunci asing (FK). Hal ini memastikan bahwa informasi pengguna seperti username dan password hanya disimpan dalam tabel yang sesuai.

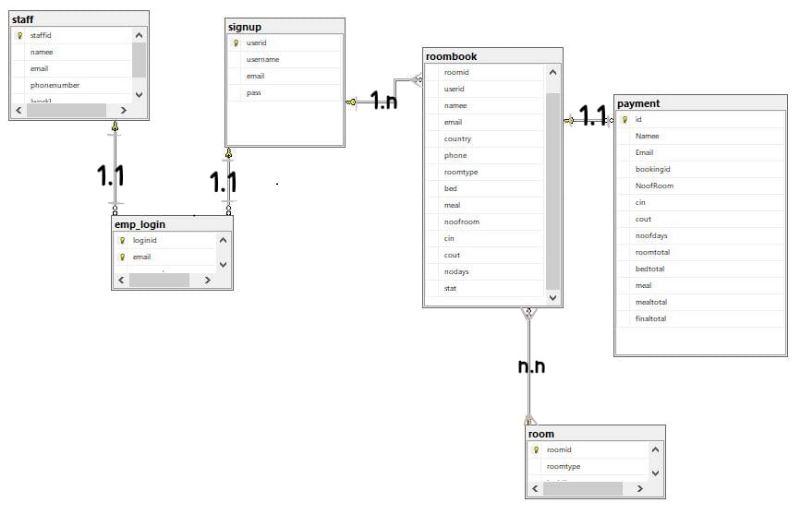
Tabel "emp\_login" tetap memenuhi 3NF setelah normalisasi. Kolom loginid berfungsi sebagai kunci primer (PK), dan kolom email dan password berisi informasi login pegawai. Kolom loginid juga berfungsi sebagai kunci asing (FK) yang merujuk ke kolom userid di tabel "signup" dan ke kolom staffid di tabel "staff" untuk menjaga integritas referensial.

Tabel "room" tidak mengalami perubahan dan memenuhi 3NF. Kolom roomid berfungsi sebagai kunci primer (PK), sedangkan kolom roomtype dan bedding berisi informasi tentang jenis kamar dan jenis tempat tidur.

Tabel "roombook" tetap memenuhi 3NF setelah normalisasi. Kolom bookingid berfungsi sebagai kunci primer (PK) yang unik untuk setiap pemesanan kamar. Kolom roomid dan userid merupakan kunci asing (FK) yang merujuk ke tabel "room" dan "signup" masing-masing. Tabel ini juga terkait dengan tabel "booking\_info" yang berisi kolom roomtype, bed, meal, noofroom, cin, cout, dan nodays yang berisi informasi terkait pemesanan kamar. Ini membantu dalam menghindari duplikasi data dan memisahkan informasi yang berkaitan dengan pemesanan kamar.

Tabel "payment" juga memenuhi 3NF setelah normalisasi. Kolom id berfungsi sebagai kunci primer (PK) yang unik untuk setiap pembayaran. Kolom bookingid merupakan kunci asing (FK) yang merujuk ke kolom bookingid di tabel "roombook" untuk menghubungkan pembayaran dengan pemesanan yang sesuai. Tabel ini juga terkait dengan tabel "roombooking" yang berisi kolom roomtotal, bedtotal, meal, dan mealtotal yang berisi informasi terkait pembayaran. Ini memisahkan informasi pembayaran dengan informasi pemesanan kamar yang terkait.

B. **Entity Relationship Diagram (ERD)**

****

**Berikut adalah penjelasan ERD (Entity-Relationship Diagram) untuk website The Sixth**

Tabel "Sign Up" menggambarkan informasi pengguna yang mendaftar di website The Sixth.

Kolom-kolom yang ada termasuk:

●  userID: ID unik untuk pengguna.

●  RoomBookID: ID unik untuk pemesanan kamar yang terkait dengan pengguna.

●  Username: Nama pengguna.

●  Email: Alamat email pengguna.

●  Password: Kata sandi pengguna.

Tabel "Emp\_login" menggambarkan informasi login untuk staf hotel di website The Sixth. Kolom-kolom yang ada termasuk:

●  empID: ID unik untuk staf hotel.

●  emp\_email: Alamat email staf hotel.

●  emp\_password: Kata sandi staf hotel.

Tabel "Staff" menggambarkan informasi staf hotel di website The Sixth. Kolom-kolom yang ada termasuk:

●  id: ID unik untuk staf hotel.

●  name: Nama staf hotel.

●  email: Alamat email staf hotel.

●  phonenumber: Nomor telepon staf hotel.

●  work: Posisi pekerjaan staf hotel.

Tabel "Room" menggambarkan informasi tentang ketersediaan kamar di website The Sixth. Kolom-kolom yang ada termasuk:

●  roomid: ID unik untuk kamar.

●  room\_type: Jenis kamar.

●  bedding: Tipe tempat tidur.

Tabel "Roombook" menggambarkan informasi pemesanan kamar oleh pengguna di website The Sixth. Kolom-kolom yang ada termasuk:

●  roomid: ID kamar yang dipesan.

●  userid: ID pengguna yang memesan.

●  namee: Nama pemesan.

●  email: Alamat email pemesan.

●  country: Negara asal pemesan.

●  phone: Nomor telepon pemesan.

●  roomtype: Jenis kamar yang dipesan.

●  bed: Tipe tempat tidur yang dipesan.

●  meal: Informasi makanan yang dipilih.

●  noofroom: Jumlah kamar yang dipesan.

●  cin: Tanggal check-in.

●  cout: Tanggal check-out.

●  nodays: Jumlah hari menginap.

●  stat: Status pemesanan.

Tabel "Payment" menggambarkan informasi pembayaran di website The Sixth. Kolom-kolom yang ada termasuk:

* namee: Nama pemesan.
* email: Alamat email pemesan.
* bookingid: ID pemesanan yang terkait dengan pembayaran.
* noofroom: Jumlah kamar yang dipesan.
* cin: Tanggal check-in.
* cout: Tanggal check-out.
* noofdays: Jumlah hari menginap.
* roomtotal: Total harga kamar.
* bedtotal: Total harga tempat tidur.
* meal: Informasi makanan yang dipilih.
* mealtotal: Total harga makanan.
* finaltotal: Total pembayaran akhir.

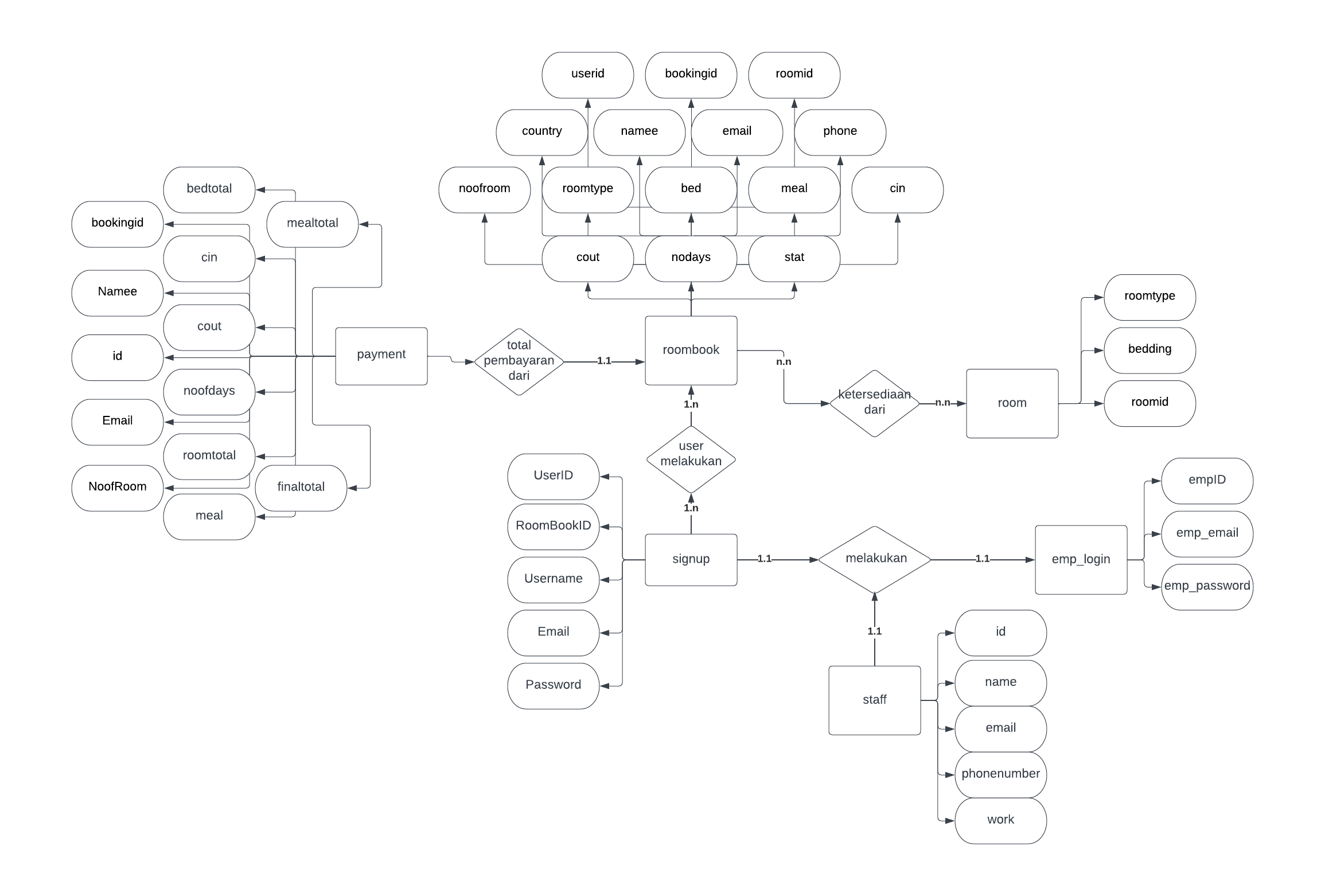
Diagram ini mencerminkan hubungan antara enititas-entitas yang terlibat dalam sistem pemesanan kamar hotel di website The Sixth. Berikut adalah penjelasan lanjutan tentang hubungan antara entitas-entitas dalam ERD diagram:   
Dalam ERD diagram ini, terdapat hubungan antara beberapa entitas yang menjelaskan bagaimana data terkait berinteraksi satu sama lain.

* 1. Tabel "Signup" memiliki hubungan one-to-one dengan tabel "Emp\_login" melalui atribut "userID" dan "empID". Hal ini menunjukkan bahwa setiap entitas dalam tabel "Signup" terkait dengan satu entitas dalam tabel "Emp\_login", dan sebaliknya. Relasi ini mengindikasikan bahwa informasi login pengguna terkait dengan informasi dan posisi pekerjaan staf hotel yang bersangkutan.
  2. Tabel "Signup" juga memiliki hubungan one-to-many dengan tabel "Roombook" melalui atribut "userID". Ini berarti bahwa satu entitas dalam tabel "Signup" dapat memiliki banyak entitas terkait dalam tabel "Roombook". Relasi ini menunjukkan bahwa pengguna yang mendaftar dapat melakukan beberapa pemesanan kamar.
  3. Tabel "Staff" memiliki hubungan one-to-one dengan tabel "Emp\_login" melalui atribut "empID". Hubungan ini menunjukkan bahwa setiap entitas dalam tabel "Staff" terkait dengan satu entitas dalam tabel "Emp\_login", dan sebaliknya. Relasi ini menggambarkan bahwa login staf hotel terkait dengan informasi dan posisi pekerjaan staf tersebut.

1. Tabel "Roombook" memiliki hubungan many-to-many dengan tabel "Room" melalui atribut "roomid". Hubungan ini menunjukkan bahwa satu pemesanan kamar dalam tabel "Roombook" dapat terkait dengan banyak entitas kamar dalam tabel "Room", dan sebaliknya. Hal ini mengindikasikan bahwa pemesanan kamar dapat terkait dengan informasi ketersediaan kamar, seperti jenis kamar dan tipe tempat tidur.
2. Tabel "Roombook" juga memiliki hubungan one-to-one dengan tabel "Payment" melalui atribut "bookingid". Ini menunjukkan bahwa setiap entitas dalam tabel "Roombook" terkait dengan satu entitas dalam tabel "Payment", dan sebaliknya. Relasi ini mengindikasikan bahwa setiap pemesanan kamar memiliki satu pembayaran yang terkait secara langsung.

Secara keseluruhan, ERD diagram ini menggambarkan hubungan antara entitas-entitas yang terlibat dalam proses reservasi dan pembayaran hotel. Informasi pengguna, pemesanan kamar, staf hotel, dan pembayaran saling terkait dan diintegrasikan melalui atribut kunci yang sesuai dalam masing-masing tabel.

**Enhanced Entity-Relationship Diagram.**



**Berikut adalah penjelasan Diagram Enhanced ERD untuk website The Sixth Dalam enhanced ERD,**

1. Tabel "signup" dan "emp login" memiliki hubungan one-to-one dengan relationship "signup (tamu) melakukan login : Dalam hubungan ini, setiap entitas "signup" (tamu) memiliki hubungan satu-satu dengan entitas "emp login". Ini berarti setiap tamu yang melakukan pendaftaran ("signup") memiliki akun login yang sesuai di tabel "emp login”.
2. Tabel "signup" memiliki hubungan one-to-many dengan "roombook" dengan relationship "signup (tamu) melakukan room booking": Dalam hubungan ini, setiap entitas "signup" (tamu) dapat melakukan beberapa pemesanan kamar ("roombook"). Namun, setiap pemesanan kamar hanya dapat terkait dengan satu entitas "signup" tertentu.
3. Tabel "roombook" dan "room" memiliki hubungan many-to-many dengan relationship "kamar dari roombooking mengambil ketersediaan kamar dari tabel kamar": Dalam hubungan ini, satu pemesanan kamar ("roombook") dapat terkait dengan banyak kamar ("room"), dan setiap kamar dapat terkait dengan beberapa pemesanan kamar. Hubungan many-to-many ini ditangani dengan menggunakan entitas tambahan, misalnya "BookingDetails", yang berfungsi sebagai entitas perantara antara "roombook" dan "room". "BookingDetails" akan menyimpan informasi tambahan seperti tanggal pemesanan, durasi, dll.
4. Tabel "roombook" dan "payment" memiliki hubungan one-to-one dengan relationship "payment mengambil total yang harus dibayar dari tabel roombook": Dalam hubungan ini, setiap entitas "roombook" hanya memiliki satu entitas "payment" yang terkait. Entitas "payment" akan mengambil informasi total yang harus dibayar dari entitas "roombook" untuk pemesanan kamar tertentu.
5. Tabel "staff" dan "emp login" memiliki hubungan one-to-one dengan relationship "staff melakukan login": Dalam hubungan ini, setiap entitas "staff" memiliki akun login yang sesuai di tabel "emp login". Ini berarti setiap staf memiliki kemampuan untuk melakukan login menggunakan kredensial yang ada dalam entitas "emp login"

2. **Menampilkan Tampilan Invoice View**

CREATE VIEW invoice\_view AS

SELECT rb\_invoice.cin AS tanggal\_invoice, rb.bookingid, rb.namee, s.email, r.roomid, rb.cin, rb.cout,

p.bedtotal, p.mealtotal, (SUM(p.finaltotal) - (SUM(p.bedtotal) + SUM(p.mealtotal))) AS Total\_room,

SUM(p.finaltotal) AS finaltotal

FROM roombook rb

JOIN room r ON rb.roomid = r.roomid

JOIN payment p ON rb.bookingid = p.bookingid

JOIN signup s ON rb.userid = s.userid

JOIN (SELECT rb.bookingid, rb.cin FROM roombook rb) rb\_invoice ON rb.bookingid = rb\_invoice.bookingid

GROUP BY r.roomid, rb.bookingid, rb.namee, s.email, rb.cin, rb.cout, p.bedtotal, p.mealtotal, rb\_invoice.cin;

Output :

A screenshot of a computer

Description automatically generated

Code di atas membuat view **invoice\_view** yang menggabungkan data dari empat tabel: **roombook**, **room**, **payment**, dan **signup**. View ini digunakan untuk menampilkan informasi invoice yang mencakup Tanggal\_Invoice bookingid, namee (nama pelanggan), email, roomid, cin (tanggal check-in), cout (tanggal check-out), bedtotal (total biaya kamar), mealtotal (total biaya makanan), Total\_room (total biaya kamar yang dihitung dengan mengurangi total pembayaran dengan jumlah biaya kamar dan biaya makanan), serta finaltotal (total pembayaran).

Pada bagian **JOIN**, dilakukan penggabungan antara tabel-tabel tersebut berdasarkan kunci yang sesuai. Fungsi JOIN digunakan untuk menghubungkan data yang memiliki nilai yang sama pada kolom **roomid (antara roombook dan room**), **bookingid (antara roombook dan payment)**, serta **userid (antara roombook dan signup)** dan mengambil kolom **bookingid** dan **cin** dari tabel **roombook**. Kemudian, join ini menghubungkan setiap entri dalam **roombook** dengan entri yang sesuai dalam subquery berdasarkan nilai yang sama pada kolom **book** . Dengan melakukan JOIN, kita dapat memperoleh data yang terkait antara tabel-tabel tersebut.

Selanjutnya, dengan menggunakan fungsi agregat SUM, kita dapat menghitung **total biaya kamar (bedtotal), total biaya makanan (mealtotal), serta total pembayaran (finaltotal)** berdasarkan **bookingid**. Pada kolom Total\_room, dilakukan perhitungan dengan mengurangi jumlah biaya kamar dan biaya makanan dari total pembayaran untuk mendapatkan total biaya kamar yang sebenarnya.

Terakhir, hasil query dikelompokkan (GROUP BY) berdasarkan bookingid, namee, email, roomid, cin, cout, bedtotal, dan mealtotal. Hal ini memastikan bahwa setiap baris data yang ditampilkan dalam view memiliki kombinasi unik dari nilai-nilai tersebut.

**Berikut penjelasan terhadap masing masing code :**



1. **CREATE VIEW invoice\_view AS**: Membuat sebuah view bernama **invoice\_view** yang akan menyimpan hasil dari query
2. **SELECT rb\_invoice.cin AS tanggal\_invoice, rb.bookingid, rb.namee, s.email, r.roomid, rb.cin, rb.cout, p.bedtotal, p.mealtotal, (SUM(p.finaltotal) - (SUM(p.bedtotal) + SUM(p.mealtotal))) AS Total\_room, SUM(p.finaltotal) AS finaltotal**: Memilih kolom-kolom yang akan ditampilkan dalam view **invoice\_view**. Beberapa kolom juga mendapatkan fungsi agregat, seperti **SUM**, untuk menghitung total bedtotal dan mealtotal, serta melakukan perhitungan Total\_room.
3. **FROM roombook rb**: Menentukan tabel utama yang akan digunakan dalam query, dengan memberikan alias **rb** ke tabel **roombook**.
4. **JOIN room r ON rb.roomid = r.roomid**: Melakukan **JOIN** antara tabel **roombook** (alias **rb**) dan tabel **room** (alias **r**) berdasarkan kolom **roomid**. Ini akan menggabungkan baris-baris yang memiliki nilai yang sama pada kolom tersebut.
5. **JOIN payment p ON rb.bookingid = p.bookingid**: Melakukan **JOIN** antara tabel **roombook** (alias **rb**) dan tabel **payment** (alias **p**) berdasarkan kolom **bookingid**. Ini akan menggabungkan baris-baris yang memiliki nilai yang sama pada kolom tersebut.
6. **JOIN signup s ON rb.userid = s.userid**: Melakukan **JOIN** antara tabel **roombook** (alias **rb**) dan tabel **signup** (alias **s**) berdasarkan kolom **userid**. Ini akan menggabungkan baris-baris yang memiliki nilai yang sama pada kolom tersebut.
7. **JOIN (SELECT rb.bookingid, rb.cin FROM roombook rb) rb\_invoice ON rb.bookingid = rb\_invoice.bookingid**: Melakukan **JOIN** dengan subquery **(SELECT rb.bookingid, rb.cin FROM roombook rb) rb\_invoice** yang menghasilkan kolom **bookingid** dan **cin** dari tabel **roombook** (alias **rb**). **JOIN** ini dilakukan berdasarkan kolom **bookingid**, sehingga menggabungkan baris-baris yang memiliki nilai yang sama pada kolom tersebut.
8. **GROUP BY r.roomid, rb.bookingid, rb.namee, s.email, rb.cin, rb.cout, p.bedtotal, p.mealtotal, rb\_invoice.cin**: Mengelompokkan hasil JOIN berdasarkan kolom-kolom yang tercantum. Hal ini diperlukan karena adanya fungsi agregat (**SUM**) pada kolom-kolom tertentu. Dengan pengelompokan ini, hasil agregat akan dihitung per grup yang terbentuk.

View "invoice\_view" yang telah dibuat sangat penting karena memberikan informasi terkait tagihan secara terkonsolidasi dan siap digunakan. Berikut adalah alasan mengapa "invoice\_view" penting:

1. **Kemudahan akses**: Dengan menggunakan "invoice\_view", pengguna dapat dengan mudah mengakses informasi yang diperlukan tentang tagihan. Semua data yang relevan telah digabungkan dalam satu tampilan yang terstruktur, sehingga pengguna tidak perlu mencari dan menggabungkan data dari beberapa tabel secara manual.
2. **Analisis dan pelaporan yang lebih efisien**: "invoice\_view" menyediakan gambaran yang komprehensif tentang tagihan, termasuk detail pemesanan, informasi pelanggan, informasi kamar, dan total tagihan. Dengan tampilan ini, pengguna dapat dengan mudah melakukan analisis dan pembuatan laporan terkait dengan keuangan, performa bisnis, atau evaluasi pengeluaran.
3. **Konsistensi data**: Dalam "invoice\_view", data yang ditampilkan berasal dari tabel-tabel terkait yang dihubungkan melalui operasi JOIN. Hal ini memastikan bahwa data yang ditampilkan dalam "invoice\_view" konsisten dan akurat, karena data diperbarui secara otomatis ketika ada perubahan atau penambahan data dalam tabel asal.
4. **Pemeliharaan yang lebih mudah:** Dengan menggunakan "invoice\_view", pemeliharaan dan pengembangan aplikasi menjadi lebih mudah. Jika ada perubahan struktur tabel atau logika pengambilan data, hanya perlu memperbarui definisi "invoice\_view" tanpa harus mengubah kode aplikasi yang menggunakan tampilan ini. Ini memudahkan pemeliharaan aplikasi dan memastikan konsistensi dalam penggunaan informasi tagihan di berbagai bagian aplikasi.
5. **Keuntungan Dari Informasi Email Pengguna dalam View "invoice\_view" untuk Manajemen Tagihan dan Komunikasi Efektif dengan Pelanggan** : Dengan adanya informasi email pengguna dalam "invoice\_view", pengguna dapat dengan mudah mengidentifikasi pelanggan yang terkait dengan setiap tagihan dan menyampaikan informasi penting kepada mereka melalui email. Hal ini memungkinkan komunikasi yang efektif antara hotel dan pelanggan, serta mempermudah manajemen tagihan dan penyelesaian transaksi.

Secara keseluruhan, "invoice\_view" membantu menyederhanakan akses dan penggunaan informasi tagihan, meningkatkan efisiensi dalam analisis dan pelaporan, serta memudahkan pemeliharaan dan pengembangan aplikasi.