

suppose you are guest: xxx or employee:xxx

Rooms

| Guest    | Function   | SQL query  | Table used                 |
|----------|--|--|----------------------------|
|          | Look up all <i>available</i> rooms (with building & wing info) | <pre>SELECT r.room_id,        r.room_number, r.room_type,        r.base_rate, r.capacity, r.floor,        r.is_smoking, w.wing_name,        b.building_name       FROM room r       JOIN wing w ON r.wing_id =                     w.wing_id       JOIN building b ON                     r.building_id = b.building_id      WHERE r.status = 'Available';</pre> | room,<br>building,<br>wing |
| Employee | Look up ALL rooms (with building & wing info)                  | <pre>SELECT r.room_id,        r.room_number, r.room_type,        r.base_rate, r.capacity, r.floor,        r.is_smoking, w.wing_name,        b.building_name       FROM room r       JOIN wing w ON r.wing_id =                     w.wing_id       JOIN building b ON                     r.building_id = b.building_id</pre>                                    | room,<br>building,<br>wing |
|          | Sort rooms (price / status / capacity / room_type)             | <pre>SELECT ... FROM room r       JOIN wing w ...       JOIN building b ...  ORDER BY r.base_rate;  ORDER BY r.status  ORDER BY r.capacity DESC  ORDER BY r.room_type</pre>  | room,<br>building,<br>wing |

## Reservations

| Guest    | Function  | SQL query  | Table used  |
|----------|---|--|---|
|          | make new reservation request<br>(this is a "fake" function) | reservation request (fake function)<br>INSERT INTO reservation_request (party_id, start_date, end_date, status, num_room, num_guest)<br>VALUES (%s, %s, %s, 'Future', %s, %s);   | reservation_request   |
|          | See own confirmed reservations                              | SELECT rr.reservation_id, rr.start_date, rr.end_date, rr.status, r.room_number, r.room_type FROM reservation_request rr JOIN room_assignment ra ON rr.reservation_id = ra.reservation_id JOIN room r ON ra.room_id = r.room_id WHERE rr.party_id = %s AND rr.status = 'Confirmed'; | reservation_request, room_assignment, room  |
|          |   |  |   |
| Employee | Look up all pending requests                                | SELECT * FROM reservation_request WHERE status = 'Future';   | reservation_request   |
|          | Assign a room (fake)  | Insert → INSERT INTO room_assignment (reservation_id, room_id, assign_date) VALUES (%s, %s, NOW());<br><br>Update → UPDATE reservation_request SET status='Confirmed' WHERE reservation_id=%s;   | reservation_requests, room_assignment   |
|          | see the room assigned                                       | SELECT rr.*, ra.room_id, r.room_number, r.room_type FROM reservation_request rr JOIN room_assignment ra ON rr.reservation_id = ra.reservation_id JOIN room r ON ra.room_id = r.room_id;  | r.room_type<br>FROM reservation_request rr JOIN room_assignment ra ON rr.reservation_id = ra.reservation_id |

|  |  |  |   |
|--|--|--|---|
|  |  |  | JOIN room r ON<br>ra.room_id =<br>r.room_id;<br>reservation_request,<br>room_assignment, room |
|--|--|--|---|

## Events

| Guest | Function                                | SQL query   | Table used  |
|-------|---|---|---|
|       | see own event                           | SELECT * FROM event<br>WHERE host_party_id = %s;  | event   |
|       | See event details (limited information) | Room use → SELECT er.room_id, r.room_number, er.date, t.name AS timeslot<br>FROM event_room_use er JOIN room r ON er.room_id = r.room_id<br>JOIN timeslot t ON er.timeslot_id = t.timeslot_id WHERE er.event_id = %s;<br>Facility use → SELECT ef.facility_id, f.capacity, ef.date, t.name AS timeslot<br>FROM event_facility_use ef JOIN facility f ON ef.facility_id = f.facility_id<br>JOIN timeslot t ON t.timeslot_id = ef.timeslot_id WHERE ef.event_id = %s; | event, event_room_use, event_facility_use, room, facility |

| Employee | Function                               | SQL query  | Table used  |
|----------|--|--|---|
|          | Full event info(host, billing, counts) | SELECT e.*, p.party_type, c.first_name, c.last_name, o.organization_name, b.status AS billing_status<br>FROM event e LEFT JOIN party p ON e.host_party_id = p.party_id LEFT JOIN customer c ON p.customer_id = c.customer_id LEFT JOIN organization o ON p.organization_id = o.organization_id LEFT JOIN billing_account b ON e.billing_account_id = b.billing_account_id; | event, party, customer, organization, billing_account |

|  |                           |   |                                      |
|--|---------------------------|---|--------------------------------------|
|  | occupied rooms/facilities | <pre>SELECT room_id FROM event_room_use WHERE date = CURDATE();  SELECT room_id FROM event_room_use WHERE date = CURDATE();</pre> | event_facility_use<br>event_room_use |
|  | available rooms           | <pre>SELECT room_id FROM event_room_use WHERE date = CURDATE(); SELECT room_id FROM event_room_use WHERE date = CURDATE();</pre>  | event_facility_use<br>event_room_use |

## Billing

| Guest | Function                    | SQL query   | Table used                                   |
|-------|-----------------------------|---|--|
|       | Look up own billing_account | SELECT * FROM<br>billing_account WHERE<br>party_id = %s;  | billing_account                              |
|       | see own charges             | SELECT c.charge_id,<br>c.total_amount,<br>c.posted_at,<br>s.description AS<br>service_desc FROM<br>charge c LEFT JOIN<br>service_usage s ON<br>c.service_usage_id =<br>s.service_usage_id<br>WHERE c.customer_id =<br>%s; | billing_account,<br>charge,<br>service_usage |

## Employee

| Guest | Function                         | SQL query  | Table used   |
|-------|----------------------------------|--|--|
|       | See all accounts + total charges | SELECT b.billing_account_id,<br>p.party_id, p.party_type,<br>SUM(c.total_amount) AS total FROM<br>billing_account b LEFT JOIN<br>charge_allocation ca ON<br>b.billing_account_id=ca.billing_account_<br>id LEFT JOIN charge c ON<br>ca.charge_id=c.charge_id LEFT JOIN<br>party p ON b.party_id=p.party_id<br>GROUP BY b.billing_account_id; | billing _acco<br>unt,<br>party,<br>charge _alloc<br>ation,<br>charge |
|       | List all charges                 | SELECT c.*, ca.responsible_percent,<br>ca.amount FROM charge c JOIN<br>charge_allocation ca ON c.charge_id =<br>ca.charge_id WHERE<br>ca.billing_account_id = %s;  | charge ,<br>charge _alloc<br>ation                                   |

Rooms

rooms\_list

● Login/Logout

login

logout

● Reservations(guest & employee)

new\_reservation

list\_reservations

confirm\_reservation ← 有参数, 不放在导航

● Events(guest & employee)

new\_event

list\_events

confirm\_event ← 有参数, 不放导航

● Facilities

list\_facilities

● Billing(guest & employee)

my\_billing

billing\_accounts

billing\_account\_detail ← 有参数, 不放导航

new\_billing\_account