# Hotel Activities Management System

## Introduction:

The Hotel Activities Management System is designed to efficiently handle and organize various activities offered by the hotel, such as swimming, gym sessions, leisure activities, and excursions. The system ensures seamless coordination and management of bookings, participations, and employee assignments, thereby enhancing guest satisfaction and operational efficiency.

## Entity Descriptions:

### Client: Represents each guest staying at the hotel.

ClientID (Primary Key): Unique identifier for each client.

Surname (varchar2): Last name of the client.

FirstName (varchar2): First name of the client.

DateOfBirth (date): Birthdate of the client.

Mail (varchar2): Email address of the client.

### Reservation: Represents the booking details for each client's stay.

- ReservationID (Primary Key) : Unique identifier for each reservation.
- ClientID (Foreign Key) : References the client making the reservation.
- HotelID (Foreign Key) : References the hotel where the reservation is made.
- CheckInDate (date) : Date when the client checks in.
- CheckOutDate (date) : Date when the client checks out.

### Activity : Represents the activities offered by the hotel.

- ActivityID (Primary Key) : Unique identifier for each activity.
- ActivityName (varchar2) : Name of the activity.

- Description (varchar2) : Description of the activity.
- StartDate (date) : Start date of the activity.
- EndDate (date) : End date of the activity.
- HotelID (Foreign Key) : References the hotel offering the activity.

## Participation : Tracks the participation of clients in activities.

- ParticipationID (Primary Key) : Unique identifier for each participation record.
- ClientID (Foreign Key) : References the client participating in the activity.
- ActivityID (Foreign Key) : References the activity in which the client participates.
- ParticipationDate (date) : Date of participation.

## Employee : Represents the staff members working at the hotel.

- EmployeeID (Primary Key) : Unique identifier for each employee.
- LastName (varchar2) : Last name of the employee.
- FirstName (varchar2) : First name of the employee.
- Position (varchar2) : Job position of the employee.
- HireDate (date) : Date when the employee was hired.
- HotelID (Foreign Key) : References the hotel where the employee works.

## Hotel : Represents the hotels in the system.

- HotelID (Primary Key) : Unique identifier for each hotel.
- Name (varchar2) : Name of the hotel.
- Address (varchar2) : Address of the hotel.
- City (varchar2) : City where the hotel is located.
- Country (varchar2) : Country where the hotel is located.

## Description of the Relationships between the Entities

Client and Reservation :

- A client can make multiple reservations. (1:N)
- Each reservation is made by a single client. (N:1)

Client and Participation :

- A client can participate in multiple activities. (1:N)
- Each participation record references a single client. (N:1)

Reservation and Hotel :

- Each reservation is for a single hotel. (N:1)
- A hotel can have multiple reservations. (1:N)

Activity and Participation :

- An activity can have multiple participation records. (1:N)
- Each participation record references a single activity. (N:1)

Employee and Hotel :

- An employee works for a single hotel. (N:1)
- A hotel employs multiple employees. (1:N)

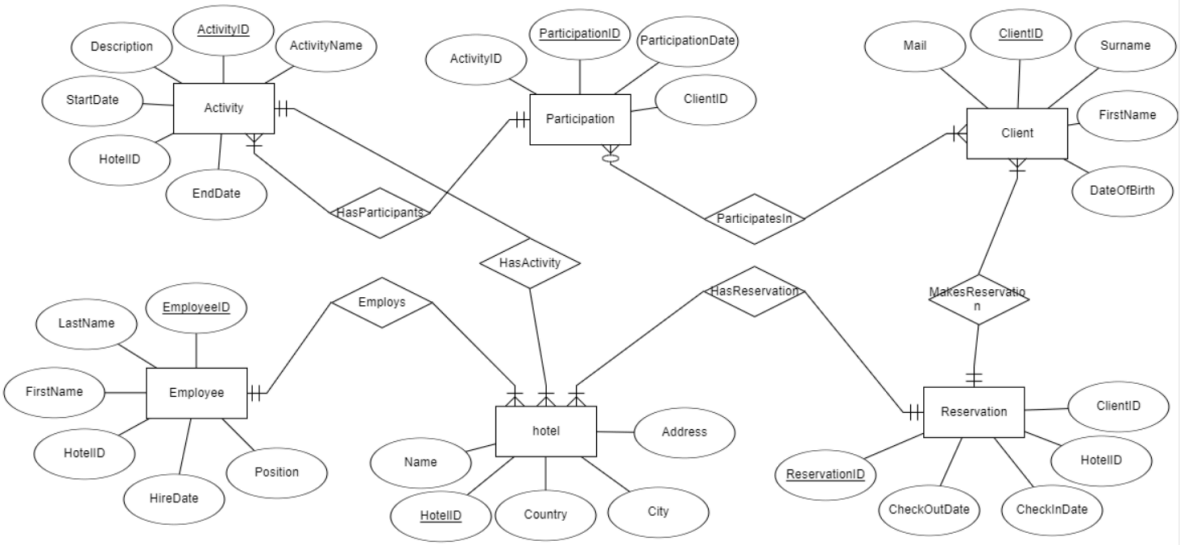
Hotel and Activity :

- A hotel can offer multiple activities. (1:N)

- Each activity is offered by a single hotel. (N:1)

These relationships ensure that the system accurately models the interconnections between clients, reservations, activities, participations, employees, and hotels.

## ERD Diagram




## Vérification de la 3NF (Troisième Forme Normale):

Pour être en 3NF, une relation doit d'abord être en 2NF, ce qui signifie qu'elle ne doit avoir aucune dépendance partielle des attributs par rapport à une clé primaire composite. Ensuite, elle doit éliminer les dépendances transitives, c'est-à-dire que les attributs non-clés ne doivent pas dépendre d'autres attributs non-clés.

1. **Hotel**
   - En 3NF : Tous les attributs dépendent uniquement de `HotelID`.
2. **Client**
   - En 3NF : Tous les attributs dépendent uniquement de `ClientID`.
3. **Reservation**
   - En 3NF : Tous les attributs dépendent uniquement de `ReservationID`, et les clés étrangères `ClientID` et `HotelID` assurent les relations avec les entités parent.
4. **Activity**
   - En 3NF : Tous les attributs dépendent uniquement de `ActivityID`, et la clé étrangère `HotelID` assure la relation avec l'entité parent.
5. **Participation**

- En 3NF : Tous les attributs dépendent uniquement de `ParticipationID`, et les clés étrangères `ActivityID` et `ClientID` assurent les relations avec les entités parent.

6. **Employee**
   - En 3NF : Tous les attributs dépendent uniquement de `EmployeeID`, et la clé étrangère `HotelID` assure la relation avec l'entité parent.

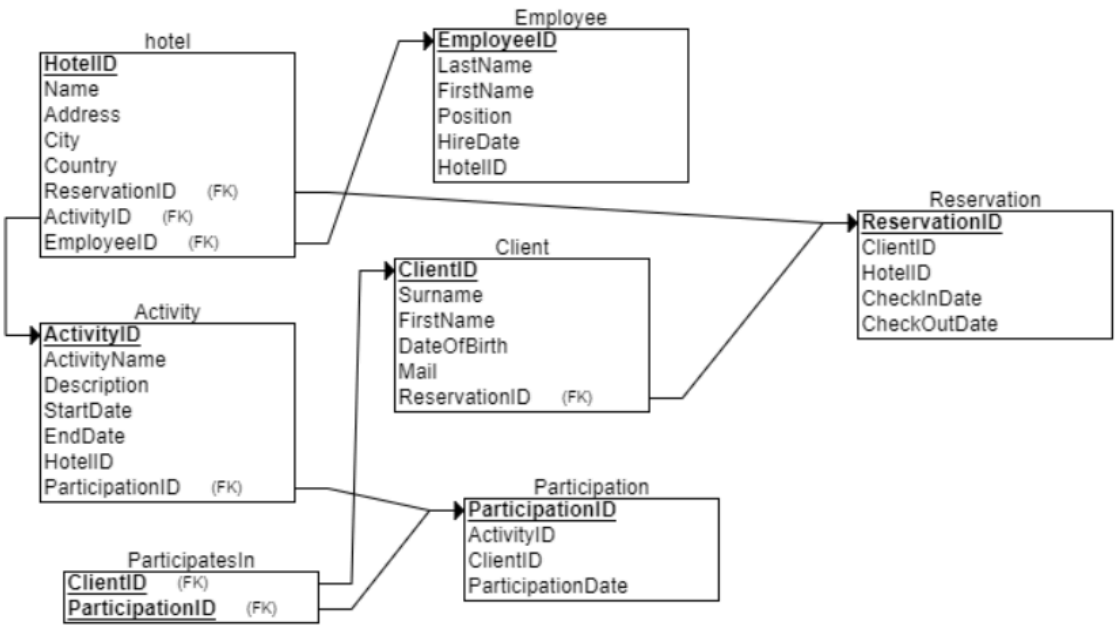
## Conclusion:

Votre diagramme ERD est bien conçu et semble correct. Toutes les entités et relations respectent les règles de la 3NF :

- Chaque attribut est directement dépendant de la clé primaire de son entité.
- Il n'y a pas de dépendances transitives entre les attributs non-clés.

Ainsi, votre diagramme est en 3NF et est cohérent avec les spécifications fournies. Vous avez bien configuré les clés primaires et les clés étrangères pour assurer l'intégrité référentielle entre les entités.

dsd:




create table:

ACTIVITY

```
-- Create table
create table ACTIVITY
(
  activityid   INTEGER not null,
  activityname VARCHAR2(50) not null,
  description  VARCHAR2(255) not null,
  startdate    DATE not null,
  enddate      DATE not null,
  hotelid      INTEGER not null
)
tablespace SYSTEM
  pctfree 10
  pctused 40
  initrans 1
  maxtrans 255;
-- Create/Recreate primary, unique and foreign key constraints
alter table ACTIVITY
  add primary key (ACTIVITYID)
  using index
  tablespace SYSTEM
  pctfree 10
  initrans 2
  maxtrans 255;
alter table ACTIVITY
  add foreign key (HOTELID)
  references HOTEL (HOTELID);
```

General | Columns | Keys | Checks | Indexes | Privileges | Triggers

Type owner  Name

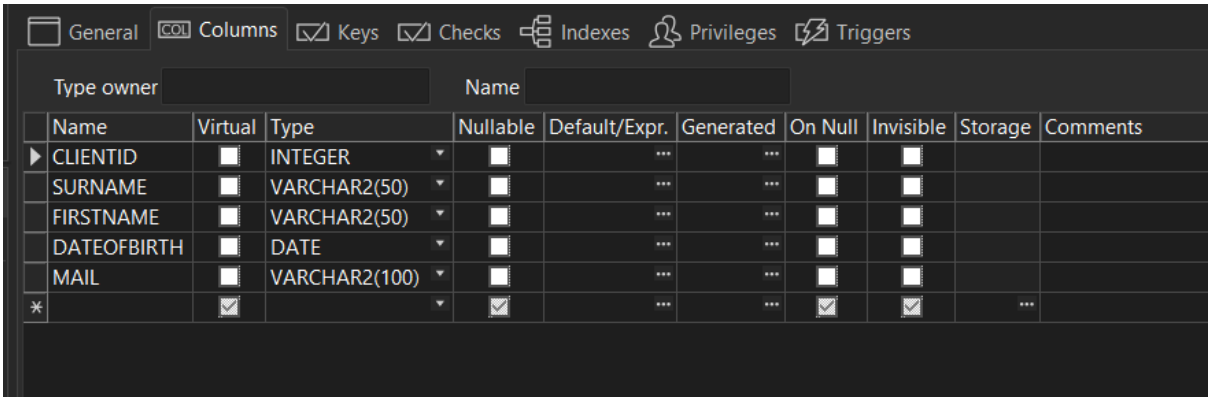
| Name | Virtual | Type | Nullable | Default/Expr. | Generated | On Null | Invisible | Storage | Comments |
|---|---|---|---|---|---|---|---|---|---|
| ACTIVITYID | ☐ | INTEGER | ☐ | ... | ... | ☐ | ☐ | | |
| ACTIVITYNAME | ☐ | VARCHAR2(50) | ☐ | ... | ... | ☐ | ☐ | | |
| DESCRIPTION | ☐ | VARCHAR2(255) | ☐ | ... | ... | ☐ | ☐ | | |
| STARTDATE | ☐ | DATE | ☐ | ... | ... | ☐ | ☐ | | |
| ENDDATE | ☐ | DATE | ☐ | ... | ... | ☐ | ☐ | | |
| HOTELID | ☐ | INTEGER | ☐ | ... | ... | ☐ | ☐ | | |
| * | ☑ | | ☑ | ... | ... | ☑ | ☑ | ... | |

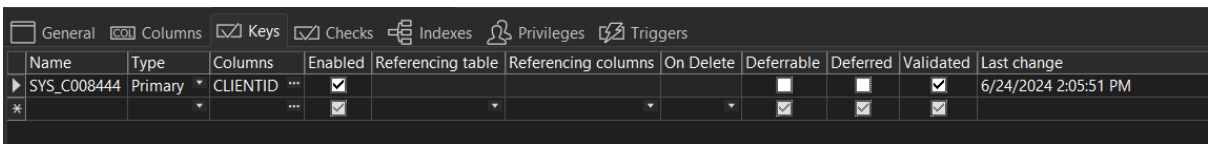
General | Columns | Keys | Checks | Indexes | Privileges | Triggers

| Name | Type | Columns | Enabled | Referencing table | Referencing columns | On Delete | Deferrable | Deferred | Validated | Last change |
|---|---|---|---|---|---|---|---|---|---|---|
| SYS_C008473 | Primary | ACTIVITYID ... | ☑ | | | | ☐ | ☐ | ☑ | 6/24/2024 2:06:17 PM |
| SYS_C008474 | Foreign | HOTELID ... | ☑ | HOTEL | HOTELID | No action | ☐ | ☐ | ☑ | 6/24/2024 2:06:17 PM |
| * | | ... | ☑ | | | | ☑ | ☑ | ☑ | |

Client:

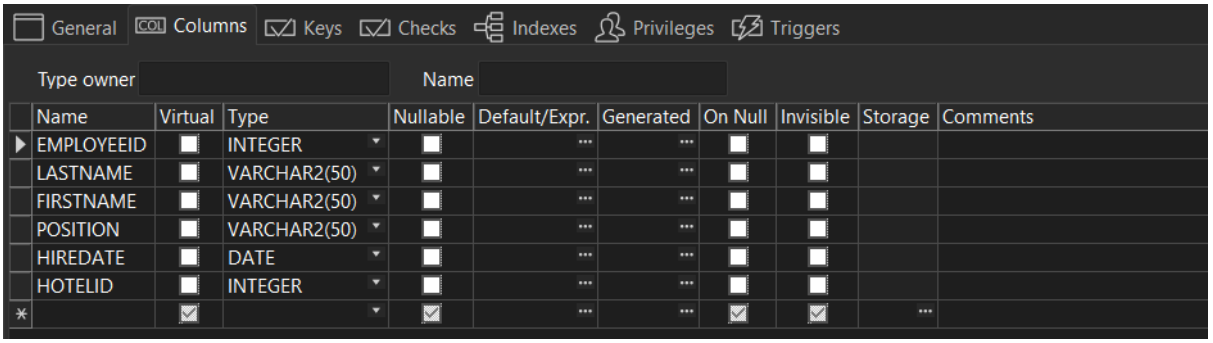
```
-- Create table
create table CLIENT
(
  clientid    INTEGER not null,
  surname     VARCHAR2(50) not null,
  firstname   VARCHAR2(50) not null,
  dateofbirth DATE not null,
  mail        VARCHAR2(100) not null
)
tablespace SYSTEM
  pctfree 10
  pctused 40
  initrans 1
  maxtrans 255;
-- Create/Recreate primary, unique and foreign key constraints
alter table CLIENT
  add primary key (CLIENTID)
  using index
  tablespace SYSTEM
  pctfree 10
  initrans 2
  maxtrans 255;
```



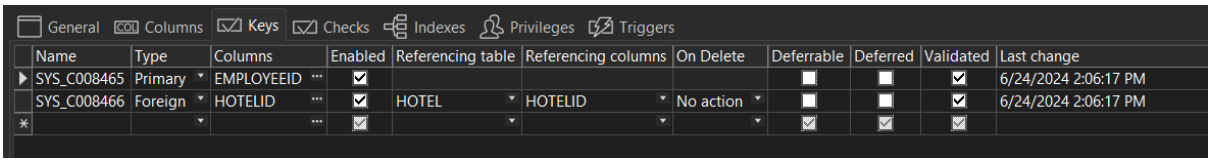



employee:

```sql
-- Create table
create table EMPLOYEE
(
  employeeid INTEGER not null,
  lastname   VARCHAR2(50) not null,
  firstname  VARCHAR2(50) not null,
  position   VARCHAR2(50) not null,
  hiredate   DATE not null,
  hotelid    INTEGER not null
)
tablespace SYSTEM
  pctfree 10
  pctused 40
  initrans 1
  maxtrans 255;
-- Create/Recreate primary, unique and foreign key constraints
alter table EMPLOYEE
  add primary key (EMPLOYEEID)
  using index
  tablespace SYSTEM
  pctfree 10
  initrans 2
  maxtrans 255;
alter table EMPLOYEE
  add foreign key (HOTELID)
  references HOTEL (HOTELID);
```



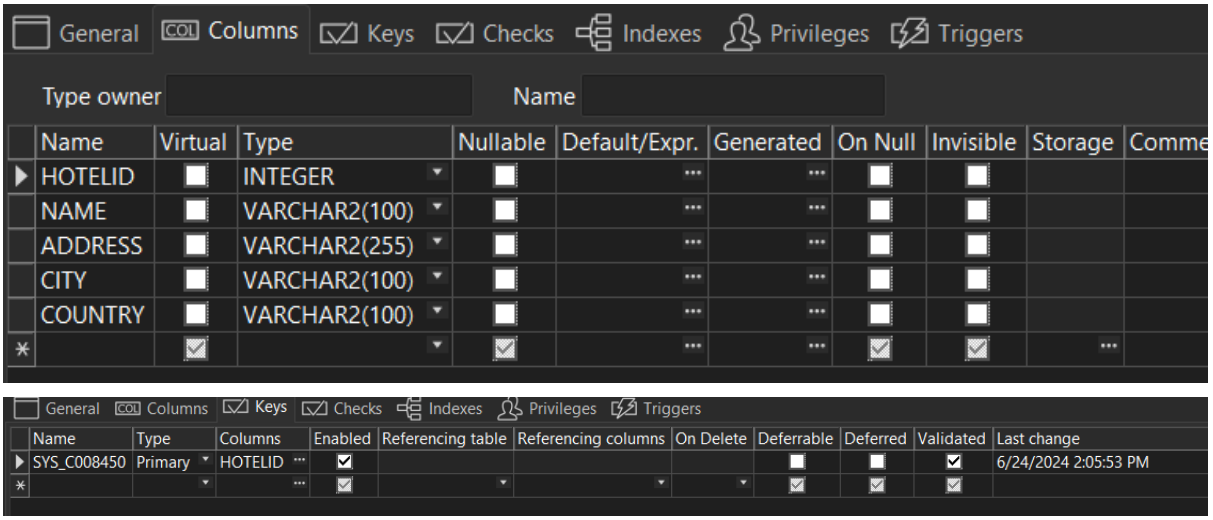

| Name | Virtual | Type | Nullable | Default/Expr. | Generated | On Null | Invisible | Storage | Comments |
|---|---|---|---|---|---|---|---|---|---|
| EMPLOYEEID | ☐ | INTEGER | ☐ | ... | ... | ☐ | ☐ | | |
| LASTNAME | ☐ | VARCHAR2(50) | ☐ | ... | ... | ☐ | ☐ | | |
| FIRSTNAME | ☐ | VARCHAR2(50) | ☐ | ... | ... | ☐ | ☐ | | |
| POSITION | ☐ | VARCHAR2(50) | ☐ | ... | ... | ☐ | ☐ | | |
| HIREDATE | ☐ | DATE | ☐ | ... | ... | ☐ | ☐ | | |
| HOTELID | ☐ | INTEGER | ☐ | ... | ... | ☐ | ☐ | | |
| * | ☑ | | ☑ | ... | ... | ☑ | ☑ | ... | |




| Name | Type | Columns | Enabled | Referencing table | Referencing columns | On Delete | Deferrable | Deferred | Validated | Last change |
|---|---|---|---|---|---|---|---|---|---|---|
| SYS_C008465 | Primary | EMPLOYEEID | ☑ | | | | ☐ | ☐ | ☑ | 6/24/2024 2:06:17 PM |
| SYS_C008466 | Foreign | HOTELID | ☑ | HOTEL | HOTELID | No action | ☐ | ☐ | ☑ | 6/24/2024 2:06:17 PM |
| * | | | ☑ | | | | ☑ | ☑ | ☑ | |

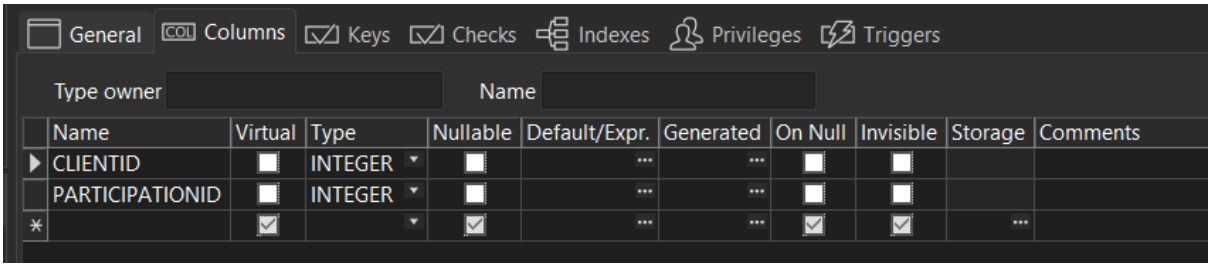
hotel:

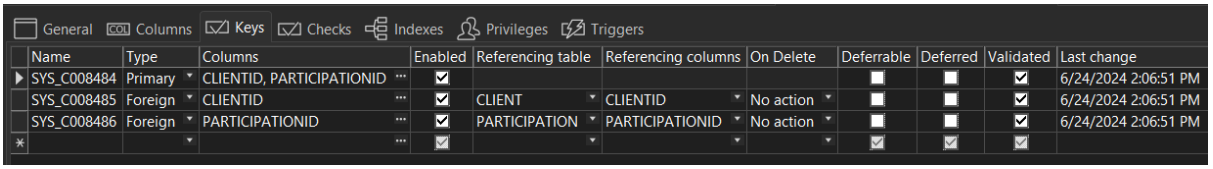
```
-- Create table
create table HOTEL
(
  hotelid INTEGER not null,
  name    VARCHAR2(100) not null,
  address VARCHAR2(255) not null,
  city    VARCHAR2(100) not null,
  country VARCHAR2(100) not null
)
tablespace SYSTEM
  pctfree 10
  pctused 40
  initrans 1
  maxtrans 255;
-- Create/Recreate primary, unique and foreign key constraints
alter table HOTEL
  add primary key (HOTELID)
  using index
  tablespace SYSTEM
  pctfree 10
  initrans 2
  maxtrans 255;
```



participationIn:

```
-- Create table
create table PARTICIPATESIN
(
  clientid        INTEGER not null,
  participationid INTEGER not null
)
tablespace SYSTEM
  pctfree 10
  pctused 40
  initrans 1
  maxtrans 255;
-- Create/Recreate primary, unique and foreign key constraints
alter table PARTICIPATESIN
  add primary key (CLIENTID, PARTICIPATIONID)
  using index
  tablespace SYSTEM
  pctfree 10
  initrans 2
  maxtrans 255;
alter table PARTICIPATESIN
  add foreign key (CLIENTID)
  references CLIENT (CLIENTID);
alter table PARTICIPATESIN
  add foreign key (PARTICIPATIONID)
  references PARTICIPATION (PARTICIPATIONID);
```

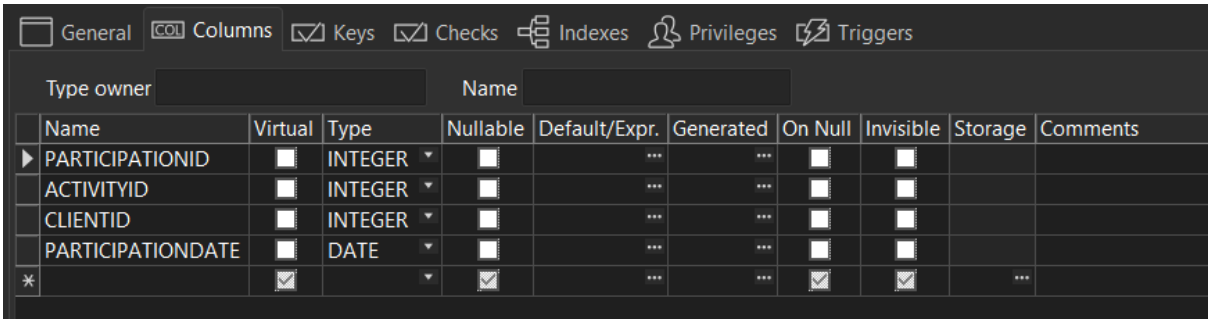


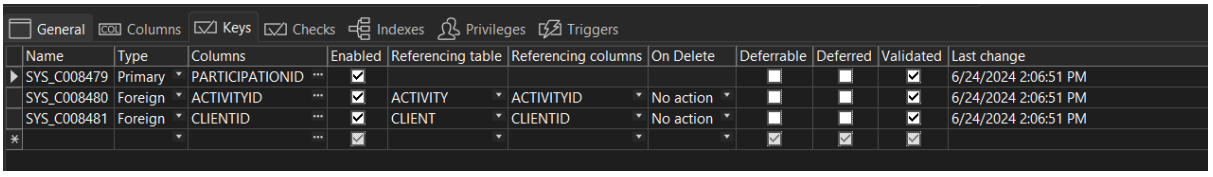



participation:

```
-- Create table
create table PARTICIPATION
(
  participationid   INTEGER not null,
  activityid        INTEGER not null,
  clientid          INTEGER not null,
  participationdate DATE not null
)
tablespace SYSTEM
  pctfree 10
  pctused 40
  initrans 1
  maxtrans 255;
-- Create/Recreate primary, unique and foreign key constraints
alter table PARTICIPATION
  add primary key (PARTICIPATIONID)
  using index
  tablespace SYSTEM
  pctfree 10
  initrans 2
  maxtrans 255;
alter table PARTICIPATION
  add foreign key (ACTIVITYID)
  references ACTIVITY (ACTIVITYID);
alter table PARTICIPATION
  add foreign key (CLIENTID)
  references CLIENT (CLIENTID);
```

Apply Refresh Close Help Query...

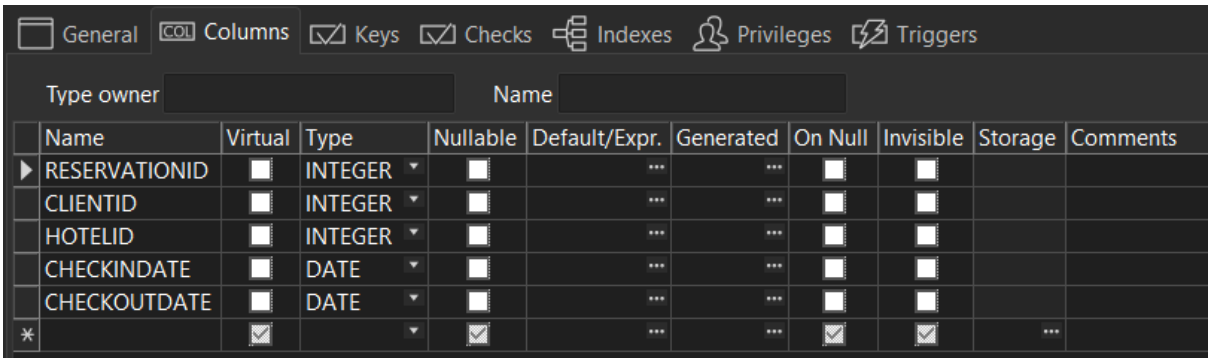


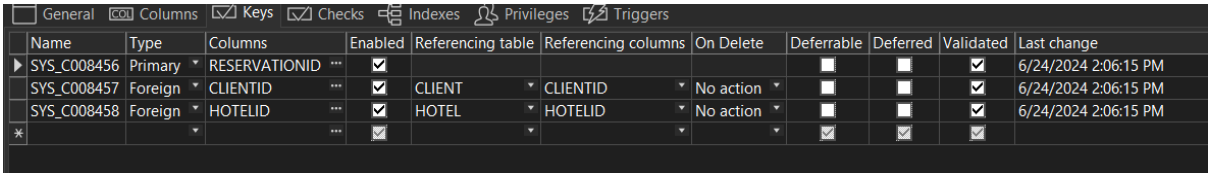



reservation:

```
-- Create table
create table RESERVATION
(
  reservationid INTEGER not null,
  clientid      INTEGER not null,
  hotelid       INTEGER not null,
  checkindate   DATE not null,
  checkoutdate  DATE not null
)
tablespace SYSTEM
  pctfree 10
  pctused 40
  initrans 1
  maxtrans 255;
-- Create/Recreate primary, unique and foreign key constraints
alter table RESERVATION
  add primary key (RESERVATIONID)
  using index
  tablespace SYSTEM
  pctfree 10
  initrans 2
  maxtrans 255;
alter table RESERVATION
  add foreign key (CLIENTID)
  references CLIENT (CLIENTID);
alter table RESERVATION
  add foreign key (HOTELID)
  references HOTEL (HOTELID);
```

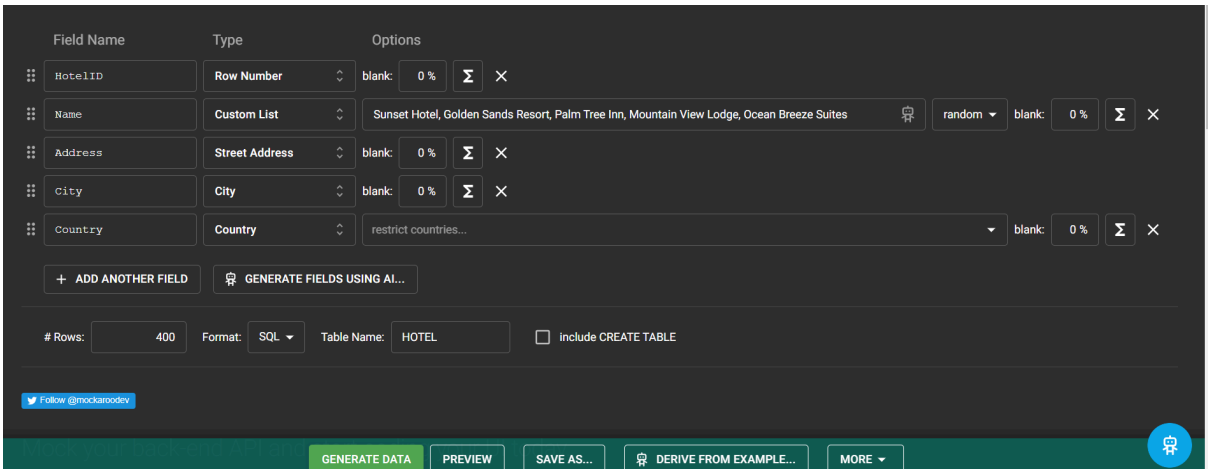
Apply Refresh Close Help Query...



## Entering data by mockaroo :

Entering data into the hotel table:



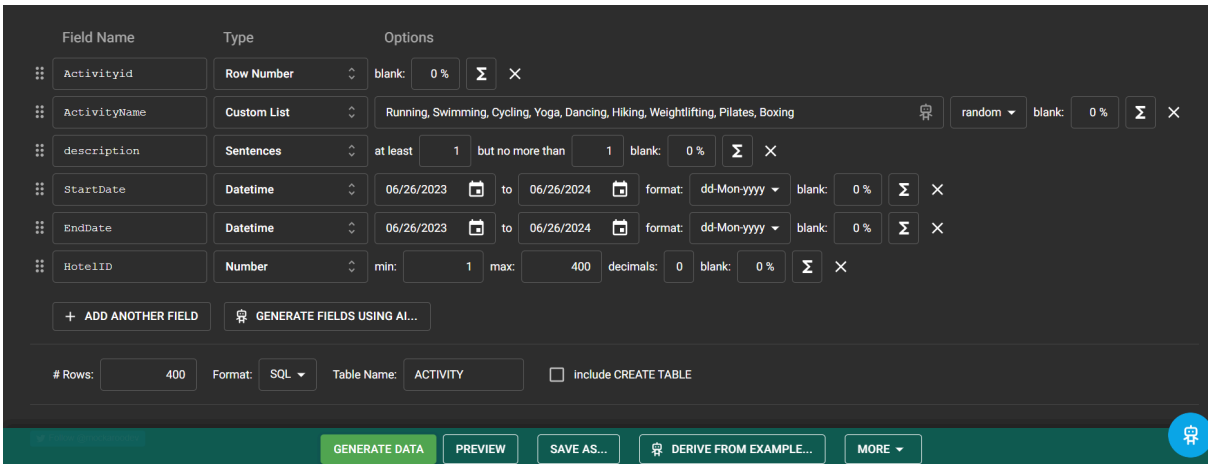

C##YEYE.HOTEL@XE | SELECT * FROM C##YEYE.HOTEL

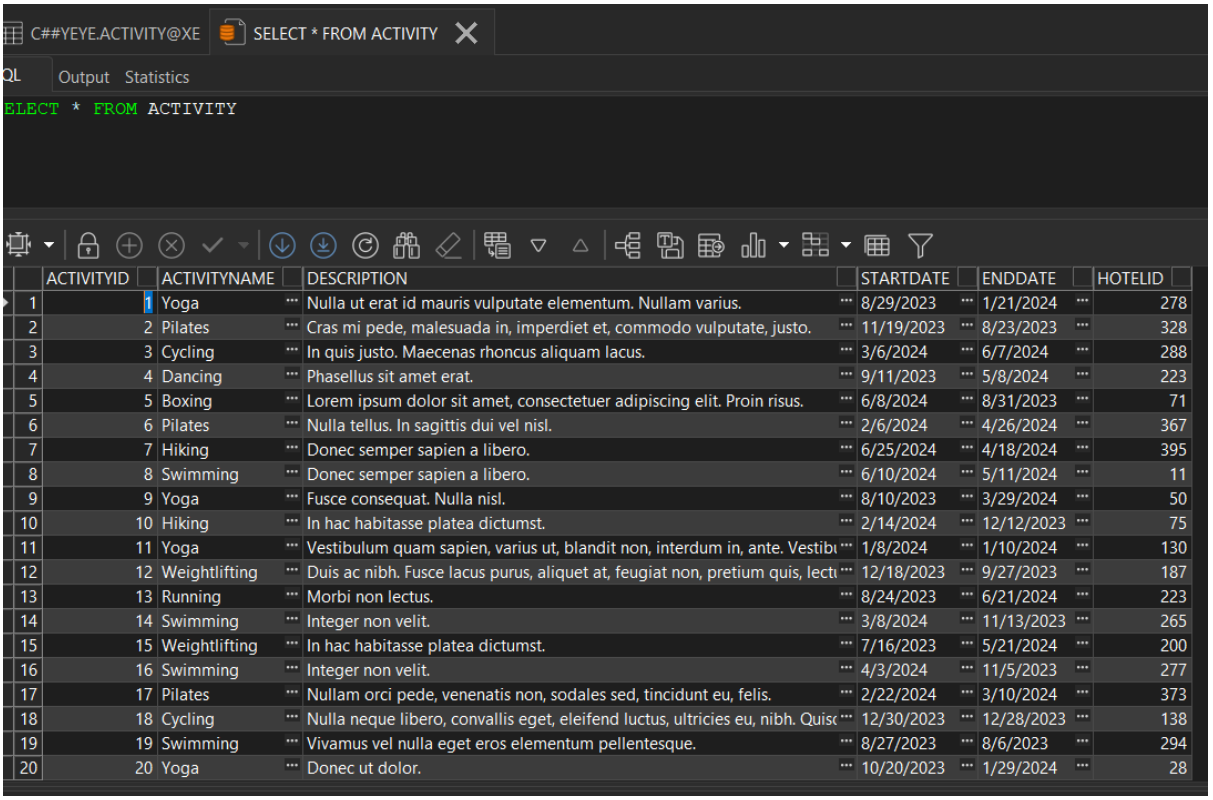
SQL | Output | Statistics

```
SELECT *
FROM C##YEYE.HOTEL
```

| | HOTELID | NAME | ADDRESS | CITY | COUNTRY |
|---|---|---|---|---|---|
| 1 | 344 | Ocean Breeze Suites | 95 Jackson Avenue | Zargaran | Afghanistan |
| 2 | 345 | Golden Sands Resort | 161 Hooker Plaza | Gampang | Indonesia |
| 3 | 346 | Sunset Hotel | 7 Bobwhite Center | Xinzhou | China |
| 4 | 347 | Palm Tree Inn | 21976 Declaration Drive | Minneapolis | United States |
| 5 | 348 | Ocean Breeze Suites | 868 1st Place | Ashiya | Japan |
| 6 | 349 | Golden Sands Resort | 43 Fordem Street | Novyye Kuz'minki | Russia |
| 7 | 350 | Mountain View Lodge | 20 Melody Center | Taketoyo | Japan |
| 8 | 351 | Ocean Breeze Suites | 16 Eliot Avenue | Youghal | Ireland |
| 9 | 352 | Palm Tree Inn | 21 Debs Court | Oslob | Philippines |
| 10 | 353 | Sunset Hotel | 8 Farwell Alley | Huotong | China |
| 11 | 354 | Sunset Hotel | 0 Melrose Alley | Vithkuq | Albania |
| 12 | 355 | Mountain View Lodge | 5137 Melvin Parkway | Ragay | Philippines |
| 13 | 356 | Ocean Breeze Suites | 4565 Texas Junction | Pingyin | China |
| 14 | 357 | Sunset Hotel | 680 Burrows Trail | Jitian | China |
| 15 | 358 | Sunset Hotel | 32 8th Lane | Frakulla e Madhe | Albania |
| 16 | 359 | Palm Tree Inn | 9 Hoard Circle | Janaúba | Brazil |
| 17 | 360 | Golden Sands Resort | 175 Granby Point | Hongwon | North Korea |
| 18 | 361 | Palm Tree Inn | 71 Dixon Junction | Los Aquijes | Peru |
| 19 | 362 | Golden Sands Resort | 869 Longview Drive | Igurubi | Tanzania |
| 20 | 363 | Ocean Breeze Suites | 67 Loomis Parkway | Ngelak | Indonesia |
| 21 | 364 | Mountain View Lodge | 38 Sloan Circle | Turi | Indonesia |

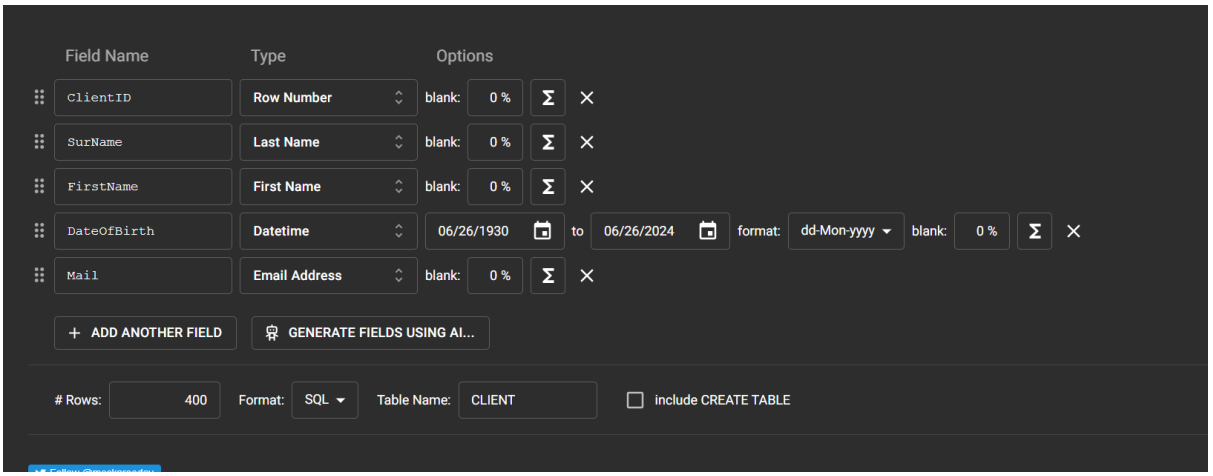
Entering data into the Activity table:

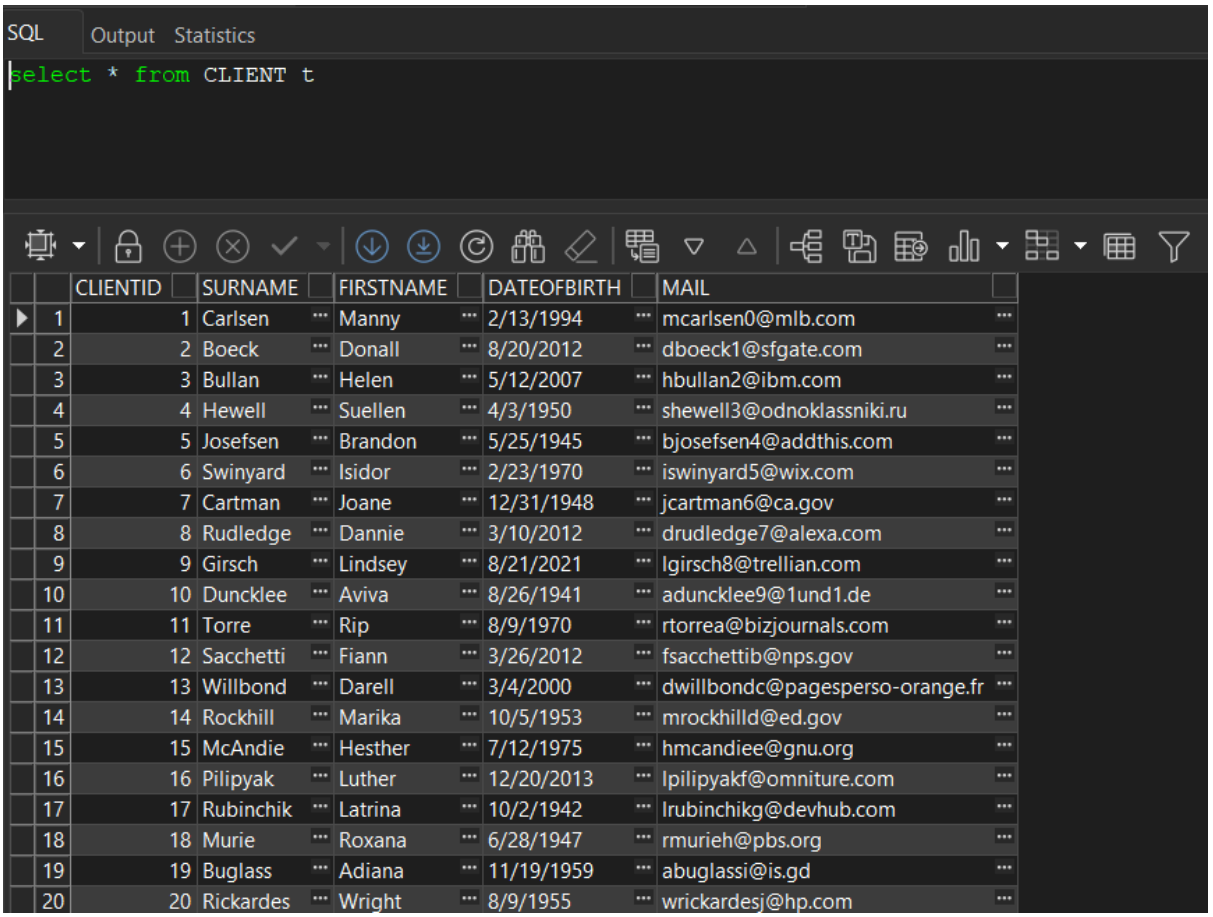



| | ACTIVITYID | ACTIVITYNAME | DESCRIPTION | STARTDATE | ENDDATE | HOTELID |
|---|---|---|---|---|---|---|
| 1 | 1 | Yoga | Nulla ut erat id mauris vulputate elementum. Nullam varius. | 8/29/2023 | 1/21/2024 | 278 |
| 2 | 2 | Pilates | Cras mi pede, malesuada in, imperdiet et, commodo vulputate, justo. | 11/19/2023 | 8/23/2023 | 328 |
| 3 | 3 | Cycling | In quis justo. Maecenas rhoncus aliquam lacus. | 3/6/2024 | 6/7/2024 | 288 |
| 4 | 4 | Dancing | Phasellus sit amet erat. | 9/11/2023 | 5/8/2024 | 223 |
| 5 | 5 | Boxing | Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Proin risus. | 6/8/2024 | 8/31/2023 | 71 |
| 6 | 6 | Pilates | Nulla tellus. In sagittis dui vel nisl. | 2/6/2024 | 4/26/2024 | 367 |
| 7 | 7 | Hiking | Donec semper sapien a libero. | 6/25/2024 | 4/18/2024 | 395 |
| 8 | 8 | Swimming | Donec semper sapien a libero. | 6/10/2024 | 5/11/2024 | 11 |
| 9 | 9 | Yoga | Fusce consequat. Nulla nisl. | 8/10/2023 | 3/29/2024 | 50 |
| 10 | 10 | Hiking | In hac habitasse platea dictumst. | 2/14/2024 | 12/12/2023 | 75 |
| 11 | 11 | Yoga | Vestibulum quam sapien, varius ut, blandit non, interdum in, ante. Vestibı | 1/8/2024 | 1/10/2024 | 130 |
| 12 | 12 | Weightlifting | Duis ac nibh. Fusce lacus purus, aliquet at, feugiat non, pretium quis, lectı | 12/18/2023 | 9/27/2023 | 187 |
| 13 | 13 | Running | Morbi non lectus. | 8/24/2023 | 6/21/2024 | 223 |
| 14 | 14 | Swimming | Integer non velit. | 3/8/2024 | 11/13/2023 | 265 |
| 15 | 15 | Weightlifting | In hac habitasse platea dictumst. | 7/16/2023 | 5/21/2024 | 200 |
| 16 | 16 | Swimming | Integer non velit. | 4/3/2024 | 11/5/2023 | 277 |
| 17 | 17 | Pilates | Nullam orci pede, venenatis non, sodales sed, tincidunt eu, felis. | 2/22/2024 | 3/10/2024 | 373 |
| 18 | 18 | Cycling | Nulla neque libero, convallis eget, eleifend luctus, ultricies eu, nibh. Quisc | 12/30/2023 | 12/28/2023 | 138 |
| 19 | 19 | Swimming | Vivamus vel nulla eget eros elementum pellentesque. | 8/27/2023 | 8/6/2023 | 294 |
| 20 | 20 | Yoga | Donec ut dolor. | 10/20/2023 | 1/29/2024 | 28 |

Entering data into the Client table:




| | CLIENTID | SURNAME | FIRSTNAME | DATEOFBIRTH | MAIL |
|---|---|---|---|---|---|
| 1 | 1 | Carlsen | Manny | 2/13/1994 | mcarlsen0@mlb.com |
| 2 | 2 | Boeck | Donall | 8/20/2012 | dboeck1@sfgate.com |
| 3 | 3 | Bullan | Helen | 5/12/2007 | hbullan2@ibm.com |
| 4 | 4 | Hewell | Suellen | 4/3/1950 | shewell3@odnoklassniki.ru |
| 5 | 5 | Josefsen | Brandon | 5/25/1945 | bjosefsen4@addthis.com |
| 6 | 6 | Swinyard | Isidor | 2/23/1970 | iswinyard5@wix.com |
| 7 | 7 | Cartman | Joane | 12/31/1948 | jcartman6@ca.gov |
| 8 | 8 | Rudledge | Dannie | 3/10/2012 | drudledge7@alexa.com |
| 9 | 9 | Girsch | Lindsey | 8/21/2021 | lgirsch8@trellian.com |
| 10 | 10 | Duncklee | Aviva | 8/26/1941 | aduncklee9@1und1.de |
| 11 | 11 | Torre | Rip | 8/9/1970 | rtorrea@bizjournals.com |
| 12 | 12 | Sacchetti | Fiann | 3/26/2012 | fsacchettib@nps.gov |
| 13 | 13 | Willbond | Darell | 3/4/2000 | dwillbondc@pagesperso-orange.fr |
| 14 | 14 | Rockhill | Marika | 10/5/1953 | mrockhilld@ed.gov |
| 15 | 15 | McAndie | Hesther | 7/12/1975 | hmcandiee@gnu.org |
| 16 | 16 | Pilipyak | Luther | 12/20/2013 | lpilipyakf@omniture.com |
| 17 | 17 | Rubinchik | Latrina | 10/2/1942 | lrubinchikg@devhub.com |
| 18 | 18 | Murie | Roxana | 6/28/1947 | rmurieh@pbs.org |
| 19 | 19 | Buglass | Adiana | 11/19/1959 | abuglassi@is.gd |
| 20 | 20 | Rickardes | Wright | 8/9/1955 | wrickardesj@hp.com |

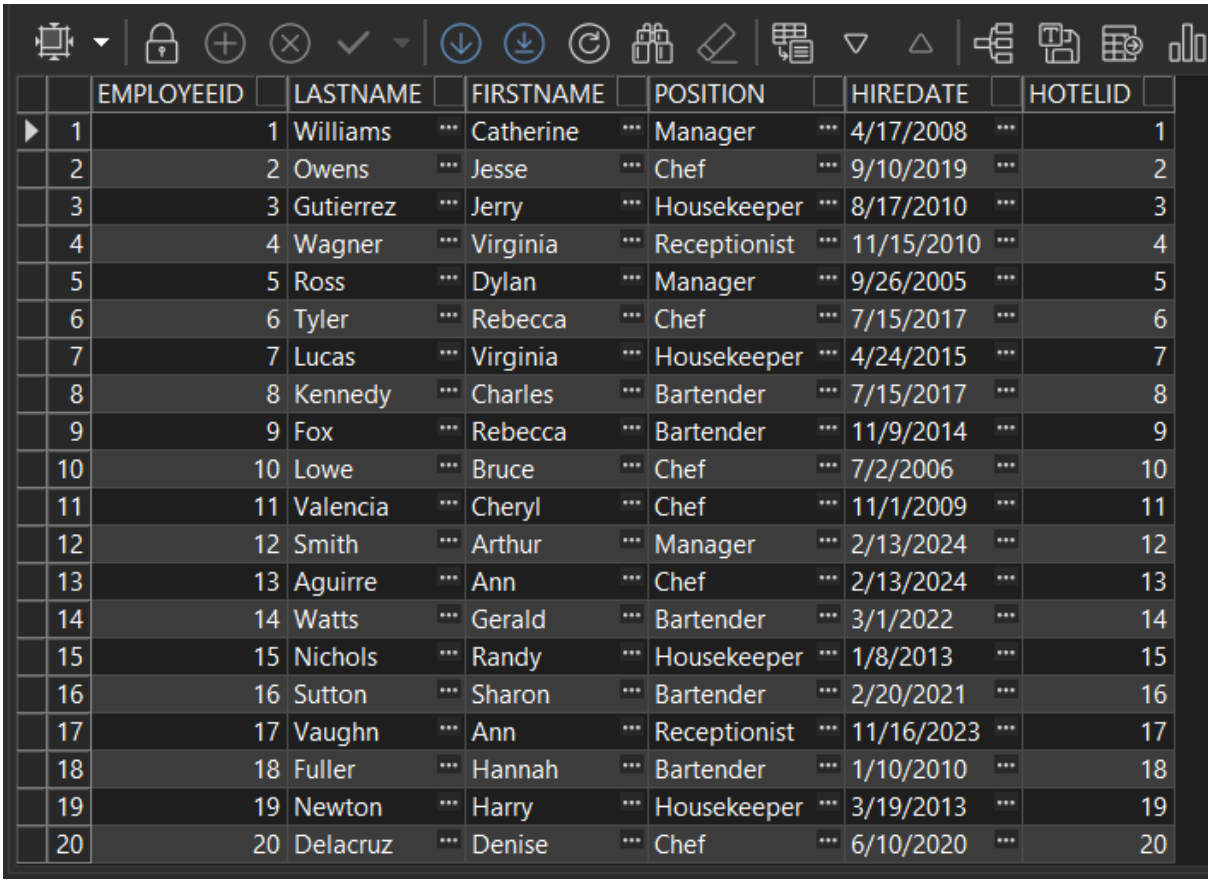
## Entering data by GENERATOR DATA.

Entering data into the employee table:

EMPLOYEE

Owner: C##YEYE | Table: EMPLOYEE | Number of records: 400..500

| Name | Type | Size | Data |
|---|---|---|---|
| EMPLOYEEID | NUMBER | | Sequence(1, 1, 400) |
| LASTNAME | VARCHAR2 | 50 | List('Smith', 'Doe', 'Brown', 'Johnson', 'Williams', 'Jones', 'Garcia', 'Miller', 'Davis', 'Rodriguez', 'Perez', 'Martinez', 'Hernandez', 'Lopez', 'Gonzalez', 'Anc |
| FIRSTNAME | VARCHAR2 | 50 | List('John', 'Jane', 'Michael', 'Emily', 'David', 'Linda', 'Richard', 'Barbara', 'James', 'Elizabeth', 'Thomas', 'Sarah', 'Robert', 'Jessica', 'Daniel', 'Nancy', 'Paul |
| POSITION | VARCHAR2 | 50 | List('Manager', 'Receptionist', 'Chef', 'Housekeeper', 'Bartender') |
| ✓ HIREDATE | DATE | | '13/02/2024', '29/03/2024', '13/05/2024', '27/06/2024', '12/08/2024', '26/09/2024', '10/11/2024', '24/12/2024') |
| HOTELID | NUMBER | | Sequence(1, 1, 400) |



| | EMPLOYEEID | LASTNAME | FIRSTNAME | POSITION | HIREDATE | HOTELID |
|---|---|---|---|---|---|---|
| 1 | 1 | Williams | Catherine | Manager | 4/17/2008 | 1 |
| 2 | 2 | Owens | Jesse | Chef | 9/10/2019 | 2 |
| 3 | 3 | Gutierrez | Jerry | Housekeeper | 8/17/2010 | 3 |
| 4 | 4 | Wagner | Virginia | Receptionist | 11/15/2010 | 4 |
| 5 | 5 | Ross | Dylan | Manager | 9/26/2005 | 5 |
| 6 | 6 | Tyler | Rebecca | Chef | 7/15/2017 | 6 |
| 7 | 7 | Lucas | Virginia | Housekeeper | 4/24/2015 | 7 |
| 8 | 8 | Kennedy | Charles | Bartender | 7/15/2017 | 8 |
| 9 | 9 | Fox | Rebecca | Bartender | 11/9/2014 | 9 |
| 10 | 10 | Lowe | Bruce | Chef | 7/2/2006 | 10 |
| 11 | 11 | Valencia | Cheryl | Chef | 11/1/2009 | 11 |
| 12 | 12 | Smith | Arthur | Manager | 2/13/2024 | 12 |
| 13 | 13 | Aguirre | Ann | Chef | 2/13/2024 | 13 |
| 14 | 14 | Watts | Gerald | Bartender | 3/1/2022 | 14 |
| 15 | 15 | Nichols | Randy | Housekeeper | 1/8/2013 | 15 |
| 16 | 16 | Sutton | Sharon | Bartender | 2/20/2021 | 16 |
| 17 | 17 | Vaughn | Ann | Receptionist | 11/16/2023 | 17 |
| 18 | 18 | Fuller | Hannah | Bartender | 1/10/2010 | 18 |
| 19 | 19 | Newton | Harry | Housekeeper | 3/19/2013 | 19 |
| 20 | 20 | Delacruz | Denise | Chef | 6/10/2020 | 20 |

Entering data into the reservation table:

RESERVATION

Owner: C##YEYE | Table: RESERVATION | Number of records: 400..500

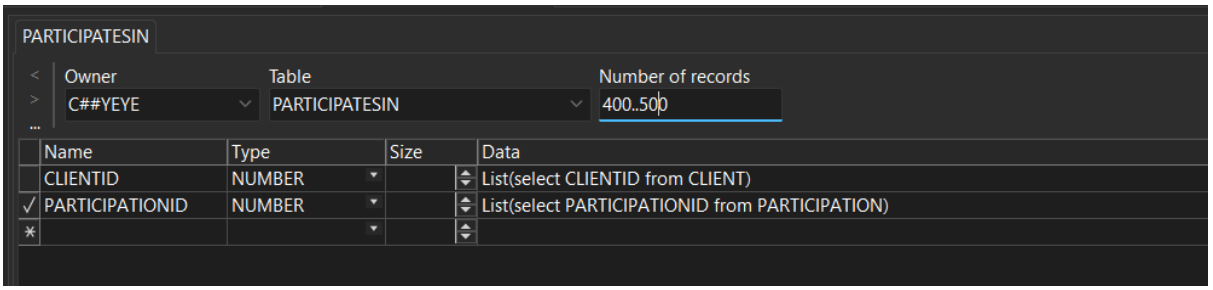
| Name | Type | Size | Data |
|---|---|---|---|
| RESERVATIONID | NUMBER | | Sequence(1000, 1, 1400) |
| CLIENTID | NUMBER | | List(select CLIENTID from CLIENT) |
| HOTELID | NUMBER | | List(select HOTELID from HOTEL) |
| CHECKINDATE | DATE | | List('01/01/2023', '15/01/2023', '30/01/2023', '14/02/2023', '28/02/2023', '15/03/2023', '30/03/2023', '14/04/2023', '28/04/2023', '14/05/2023', '28/0 |
| CHECKOUTDATE | DATE | | List('01/01/2024', '15/01/2024', '30/01/2024', '14/02/2024', '28/02/2024', '15/03/2024', '30/03/2024', '14/04/2024', '28/04/2024', '14/05/2024', '28/0 |

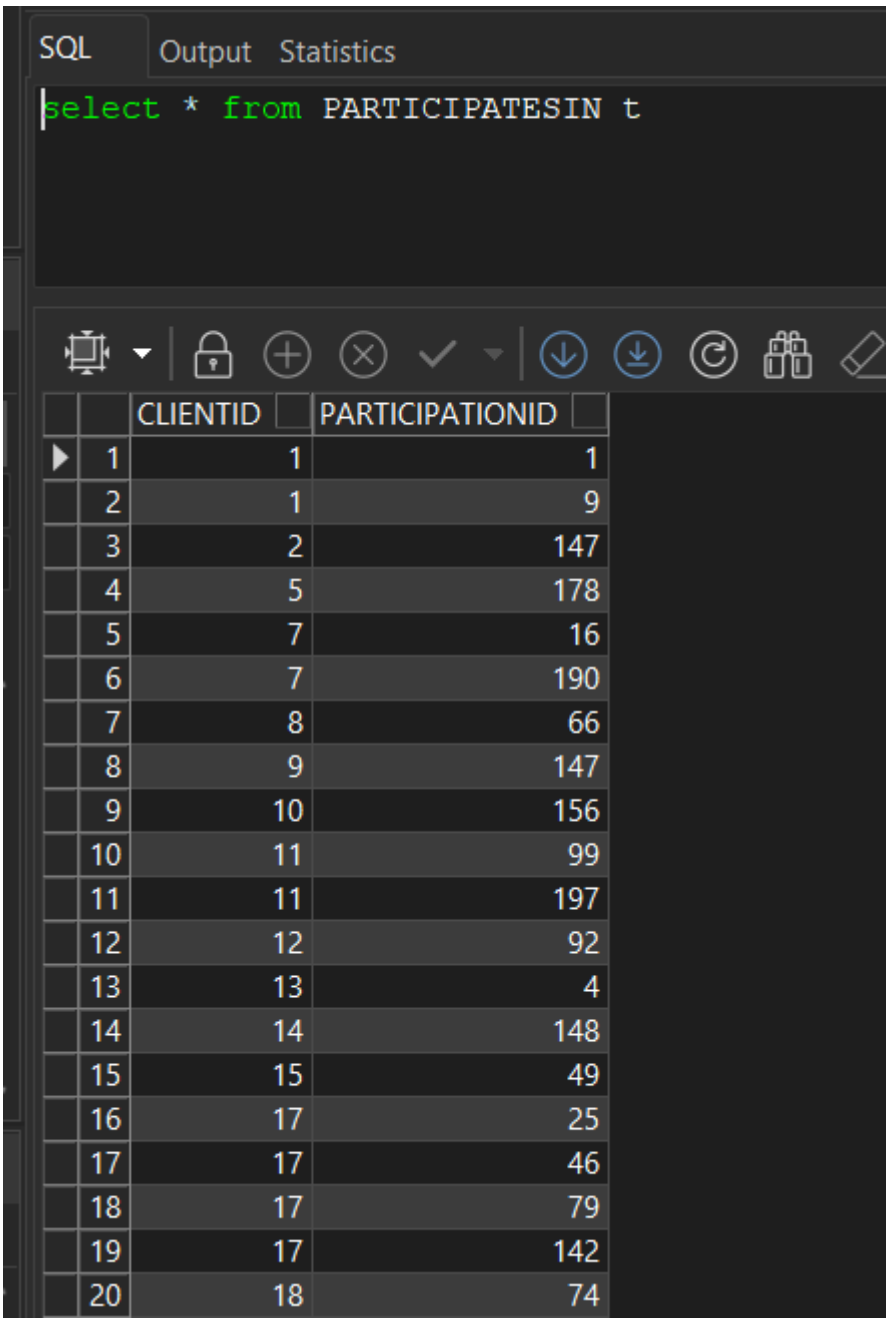
SQL Output Statistics

```
select * from RESERVATION t
```

| | RESERVATIONID | CLIENTID | HOTELID | CHECKINDATE | CHECKOUTDATE |
|---|---|---|---|---|---|
| 1 | 1000 | 377 | 193 | 11/15/2023 | 1/31/2024 |
| 2 | 1001 | 315 | 288 | 3/15/2023 | 3/30/2024 |
| 3 | 1002 | 264 | 121 | 6/28/2023 | 3/31/2024 |
| 4 | 1003 | 104 | 211 | 4/14/2023 | 1/1/2024 |
| 5 | 1004 | 271 | 238 | 4/14/2023 | 12/28/2024 |
| 6 | 1005 | 138 | 215 | 5/16/2023 | 3/15/2024 |
| 7 | 1006 | 17 | 18 | 1/7/2023 | 7/28/2024 |
| 8 | 1007 | 50 | 70 | 10/31/2023 | 8/14/2024 |
| 9 | 1008 | 27 | 345 | 10/14/2023 | 9/14/2024 |
| 10 | 1009 | 327 | 133 | 9/28/2023 | 11/15/2024 |
| 11 | 1010 | 236 | 293 | 9/16/2023 | 2/1/2024 |
| 12 | 1011 | 10 | 263 | 11/28/2023 | 4/15/2024 |
| 13 | 1012 | 360 | 151 | 5/14/2023 | 5/28/2024 |
| 14 | 1013 | 322 | 356 | 2/14/2023 | 2/28/2024 |
| 15 | 1014 | 182 | 72 | 9/16/2023 | 1/1/2024 |
| 16 | 1015 | 222 | 104 | 1/5/2023 | 10/16/2024 |
| 17 | 1016 | 166 | 349 | 1/1/2023 | 9/16/2024 |
| 18 | 1017 | 280 | 386 | 1/31/2023 | 9/28/2024 |
| 19 | 1018 | 2 | 113 | 12/14/2023 | 1/31/2024 |
| 20 | 1019 | 80 | 278 | 1/3/2023 | 7/31/2024 |

Entering data into the participateIn table:

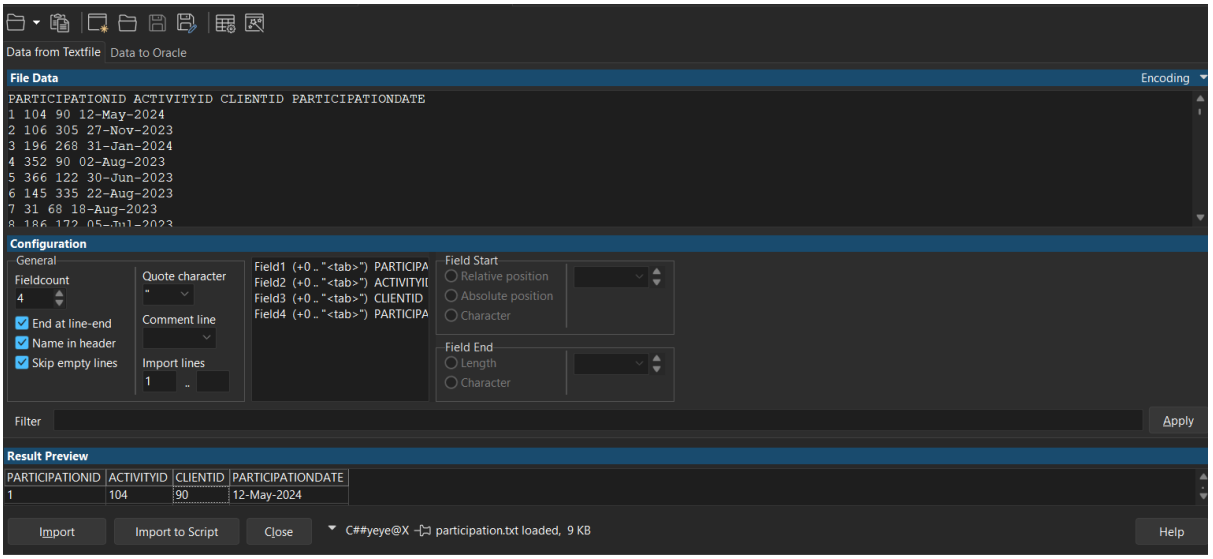


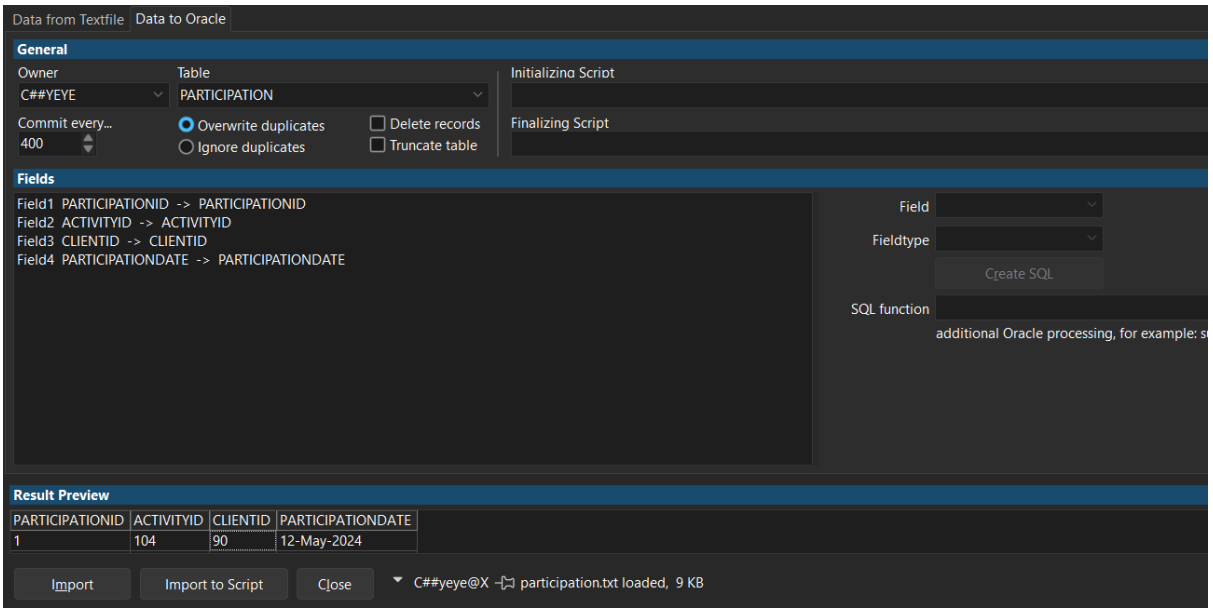




| | CLIENTID | PARTICIPATIONID |
|---|---|---|
| 1 | 1 | 1 |
| 2 | 1 | 9 |
| 3 | 2 | 147 |
| 4 | 5 | 178 |
| 5 | 7 | 16 |
| 6 | 7 | 190 |
| 7 | 8 | 66 |
| 8 | 9 | 147 |
| 9 | 10 | 156 |
| 10 | 11 | 99 |
| 11 | 11 | 197 |
| 12 | 12 | 92 |
| 13 | 13 | 4 |
| 14 | 14 | 148 |
| 15 | 15 | 49 |
| 16 | 17 | 25 |
| 17 | 17 | 46 |
| 18 | 17 | 79 |
| 19 | 17 | 142 |
| 20 | 18 | 74 |

## Entering data by text file.

Entering data into the participation table:



SQL Output Statistics

```
select * from PARTICIPATION t
```

| | PARTICIPATIONID | ACTIVITYID | CLIENTID | PARTICIPATIONDATE |
|---|---|---|---|---|
| 1 | 1 | 104 | 90 | 5/12/2024 |
| 2 | 2 | 106 | 305 | 11/27/2023 |
| 3 | 4 | 352 | 90 | 8/2/2023 |
| 4 | 5 | 366 | 122 | 6/30/2023 |
| 5 | 6 | 145 | 335 | 8/22/2023 |
| 6 | 7 | 31 | 68 | 8/18/2023 |
| 7 | 9 | 351 | 76 | 8/5/2023 |
| 8 | 10 | 135 | 211 | 9/6/2023 |
| 9 | 11 | 354 | 143 | 6/18/2024 |
| 10 | 15 | 49 | 134 | 1/26/2024 |
| 11 | 16 | 13 | 160 | 2/25/2024 |
| 12 | 17 | 162 | 338 | 1/16/2024 |
| 13 | 21 | 177 | 64 | 2/14/2024 |
| 14 | 22 | 64 | 199 | 3/8/2024 |
| 15 | 23 | 25 | 360 | 6/20/2024 |
| 16 | 24 | 399 | 35 | 2/11/2024 |
| 17 | 25 | 136 | 243 | 6/9/2024 |
| 18 | 26 | 25 | 3 | 6/26/2023 |
| 19 | 28 | 97 | 78 | 7/21/2023 |
| 20 | 29 | 56 | 335 | 6/8/2024 |

## שלב ב : שאילתות ואילוצים.

# select:

1. Selecting all participation for a specific client

```
SQL  Output  Statistics
SELECT p.PARTICIPATIONID, p.ACTIVITYID, p.CLIENTID, p.PARTICIPATIONDATE, c.FIRSTNAME, c.SURNAME
FROM PARTICIPATION p
JOIN CLIENT c ON p.CLIENTID = c.CLIENTID
WHERE p.CLIENTID = 90;
```

| | PARTICIPATIONID | ACTIVITYID | CLIENTID | PARTICIPATIONDATE | FIRSTNAME | SURNAME |
|---|---|---|---|---|---|---|
| 1 | 1 | 104 | 90 | 5/12/2024 | Jayme | Brabbs |
| 2 | 4 | 352 | 90 | 8/2/2023 | Jayme | Brabbs |
| 3 | 61 | 69 | 90 | 10/23/2023 | Jayme | Brabbs |

2. election of all participation for a specific activity

```
SQL  Output  Statistics
SELECT p.PARTICIPATIONID, p.ACTIVITYID, p.CLIENTID, p.PARTICIPATIONDATE, a.ACTIVITYNAME
FROM PARTICIPATION p
JOIN ACTIVITY a ON p.ACTIVITYID = a.ACTIVITYID
WHERE p.ACTIVITYID = 10;
```

| | PARTICIPATIONID | ACTIVITYID | CLIENTID | PARTICIPATIONDATE | ACTIVITYNAME |
|---|---|---|---|---|---|
| 1 | 101 | 10 | 295 | 8/1/2023 | Hiking |
| 2 | 152 | 10 | 107 | 1/14/2024 | Hiking |

3. Selection of participation by month and year

```
SQL   Output   Statistics
SELECT EXTRACT(YEAR FROM p.PARTICIPATIONDATE) AS Year, EXTRACT(MONTH FROM p.PARTICIPATIONDATE) AS Month, COUNT(*) AS ParticipationCount
FROM PARTICIPATION p
GROUP BY EXTRACT(YEAR FROM p.PARTICIPATIONDATE), EXTRACT(MONTH FROM p.PARTICIPATIONDATE)
ORDER BY Year, Month;
```

| | YEAR | MONTH | PARTICIPATIONCOUNT |
|---|---|---|---|
| 1 | 2023 | 6 | 3 |
| 2 | 2023 | 7 | 8 |
| 3 | 2023 | 8 | 16 |
| 4 | 2023 | 9 | 10 |
| 5 | 2023 | 10 | 5 |
| 6 | 2023 | 11 | 4 |
| 7 | 2023 | 12 | 9 |
| 8 | 2024 | 1 | 9 |
| 9 | 2024 | 2 | 6 |
| 10 | 2024 | 3 | 9 |
| 11 | 2024 | 4 | 9 |
| 12 | 2024 | 5 | 7 |
| 13 | 2024 | 6 | 11 |

4. Selection of clients with the number of participations

```
SELECT c.CLIENTID, c.FIRSTNAME, c.SURNAME, COUNT(p.PARTICIPATIONID) AS ParticipationCount
FROM CLIENT c
JOIN PARTICIPATION p ON c.CLIENTID = p.CLIENTID
GROUP BY c.CLIENTID, c.FIRSTNAME, c.SURNAME
ORDER BY ParticipationCount DESC;
```

| | CLIENTID | FIRSTNAME | SURNAME | PARTICIPATIONCOUNT |
|---|---|---|---|---|
| 1 | 90 | Jayme | Brabbs | 3 |
| 2 | 35 | Toby | Llewellyn | 3 |
| 3 | 96 | Logan | Blockley | 3 |
| 4 | 338 | Grace | Stammers | 2 |
| 5 | 239 | Nola | Granville | 2 |
| 6 | 272 | Garv | Sailer | 2 |
| 7 | 296 | Fenelia | Pesterfield | 2 |
| 8 | 314 | Jodi | Ivanichev | 2 |
| 9 | 321 | Ansley | Niezen | 2 |
| 10 | 335 | Rand | Nazaret | 2 |
| 11 | 158 | Aviva | Alker | 2 |
| 12 | 78 | Shandra | Bernhardt | 2 |
| 13 | 64 | Tobi | Bere | 2 |
| 14 | 178 | Aleda | Dymick | 2 |
| 15 | 3 | Helen | Bullan | 1 |
| 16 | 14 | Marika | Rockhill | 1 |
| 17 | 21 | Toddie | Joreau | 1 |

# Delete:

1. Deleting participation before a specific date

```
DELETE FROM PARTICIPATION
WHERE PARTICIPATIONDATE < TO_DATE('01/01/2023', 'DD/MM/YYYY');
```

```
2:63    C##yeye@XE [9:52:18 AM]  0 rows deleted in 0.006 seconds
```

2. Deleting participations for a specific activity

```
DELETE FROM PARTICIPATION
WHERE ACTIVITYID = 104;
```

# Update:

1. Updating the join date for a specific client

```
UPDATE PARTICIPATION
SET PARTICIPATIONDATE = TO_DATE('01/01/2024', 'DD/MM/YYYY')
WHERE CLIENTID = 90;
```

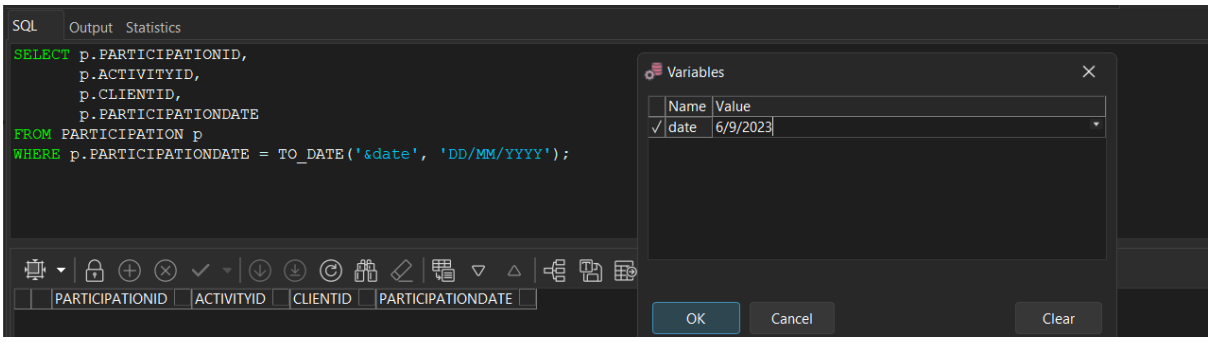
2. Updating the activity ID for a specific participation

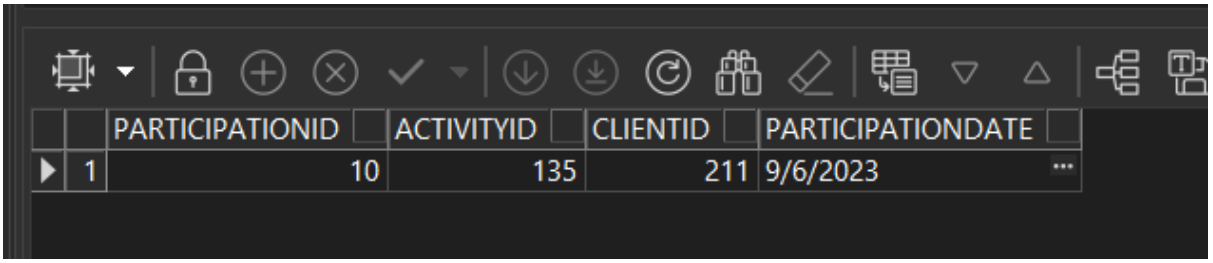
SQL | Output | Statistics

```
UPDATE PARTICIPATION
SET ACTIVITYID = 105
WHERE PARTICIPATIONID = 1;
```

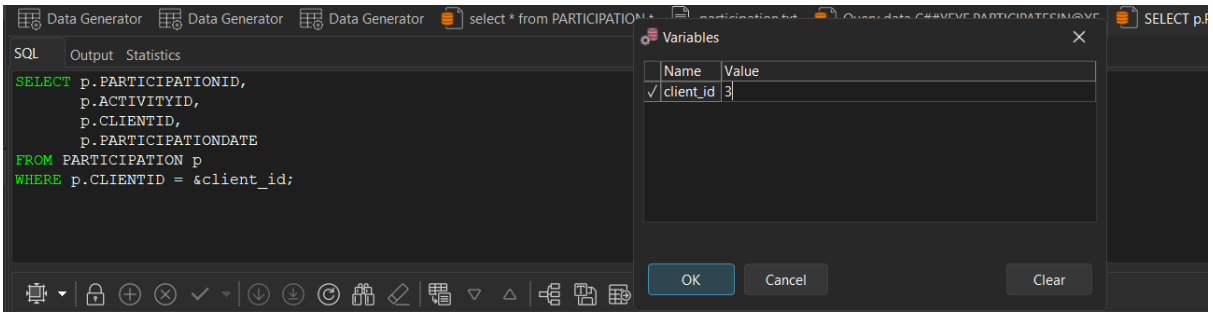
# ParamsQueries:

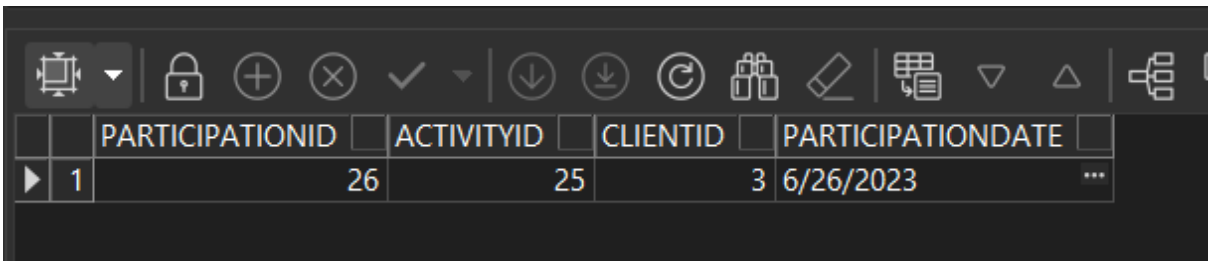
1. Selecting all participation for a specific date



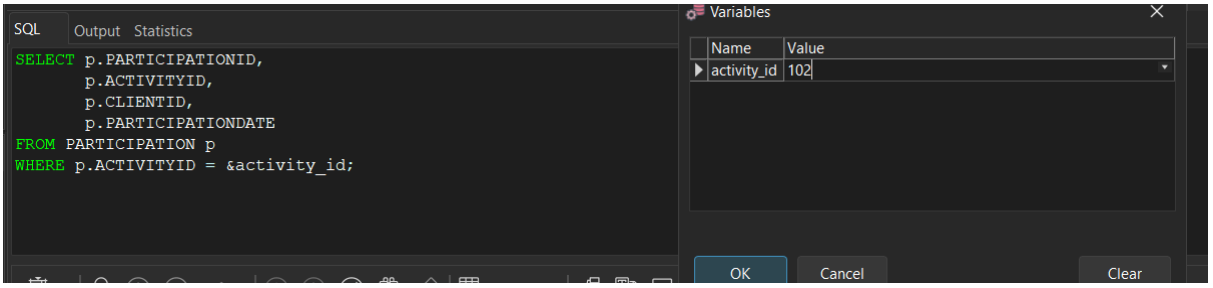

2. Selecting all entries for a specific client

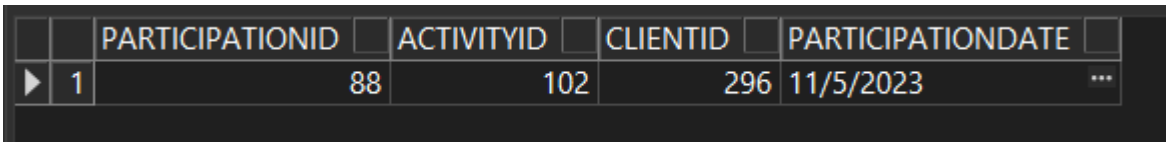




| | PARTICIPATIONID | ACTIVITYID | CLIENTID | PARTICIPATIONDATE |
|---|---|---|---|---|
| 1 | 26 | 25 | 3 | 6/26/2023 |

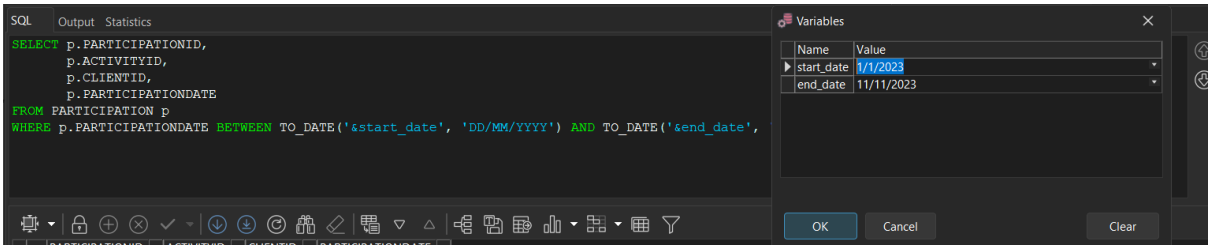
3. Selecting all entries for a specific activity

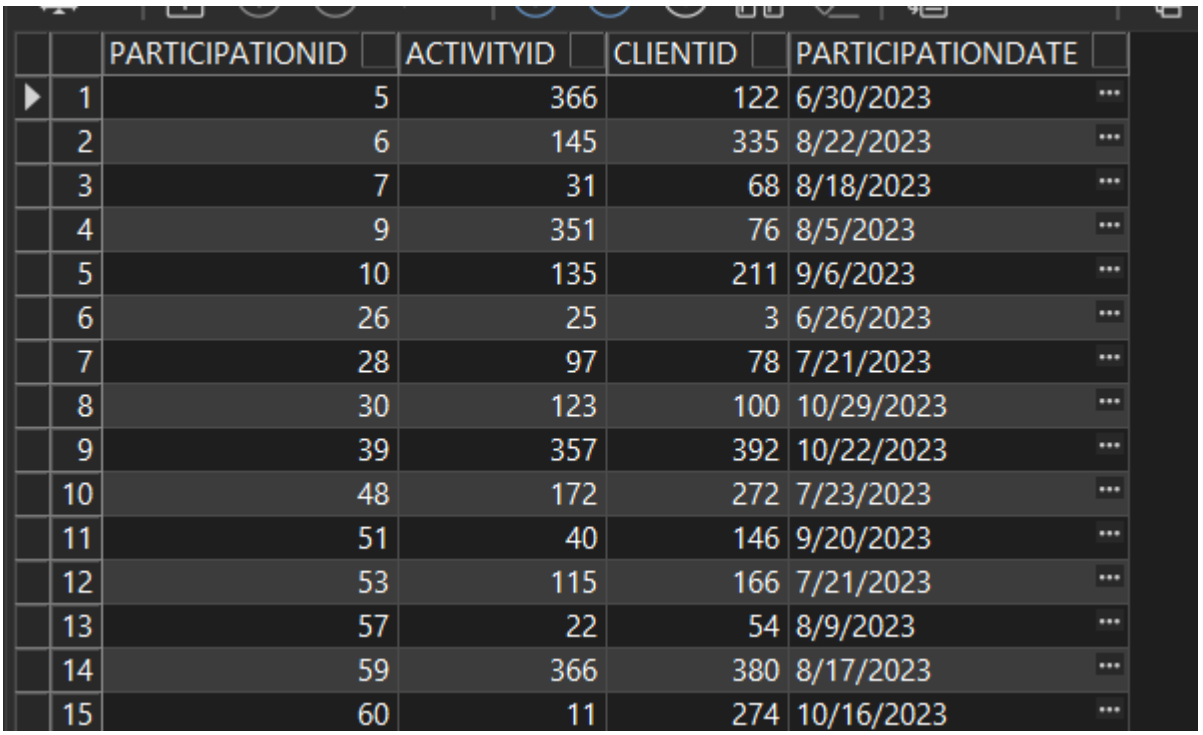


| | PARTICIPATIONID | ACTIVITYID | CLIENTID | PARTICIPATIONDATE |
|---|---|---|---|---|
| 1 | 88 | 102 | 296 | 11/5/2023 |

4. Selection of entries between two dates





| | PARTICIPATIONID | ACTIVITYID | CLIENTID | PARTICIPATIONDATE |
|---|---|---|---|---|
| 1 | 5 | 366 | 122 | 6/30/2023 |
| 2 | 6 | 145 | 335 | 8/22/2023 |
| 3 | 7 | 31 | 68 | 8/18/2023 |
| 4 | 9 | 351 | 76 | 8/5/2023 |
| 5 | 10 | 135 | 211 | 9/6/2023 |
| 6 | 26 | 25 | 3 | 6/26/2023 |
| 7 | 28 | 97 | 78 | 7/21/2023 |
| 8 | 30 | 123 | 100 | 10/29/2023 |
| 9 | 39 | 357 | 392 | 10/22/2023 |
| 10 | 48 | 172 | 272 | 7/23/2023 |
| 11 | 51 | 40 | 146 | 9/20/2023 |
| 12 | 53 | 115 | 166 | 7/21/2023 |
| 13 | 57 | 22 | 54 | 8/9/2023 |
| 14 | 59 | 366 | 380 | 8/17/2023 |
| 15 | 60 | 11 | 274 | 10/16/2023 |

# Constraints:

1. Client ID is positive

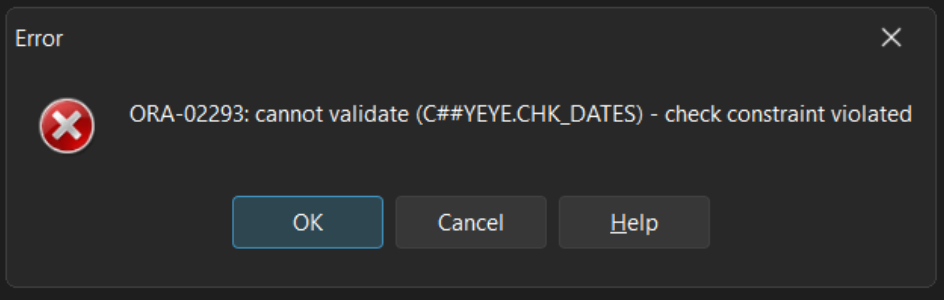
```
ALTER TABLE PARTICIPATION
ADD CONSTRAINT chk_clientid CHECK (CLIENTID > 0);
```

2. name of activity not null

```
ALTER TABLE ACTIVITY
ADD CONSTRAINT chk_activityname CHECK (ACTIVITYNAME IS NOT NULL AND ACTIVITYNAME <> '');
```

3. the departure date is later than the arrival date and the duration of the reservation does not exceed 30 days.

```
ALTER TABLE RESERVATION
ADD CONSTRAINT chk_dates CHECK (
    CHECKOUTDATE > CHECKINDATE
    AND CHECKOUTDATE <= CHECKINDATE + INTERVAL '30' DAY
);
```



# שלב ג : תכנות

## Main1 - Program Description

Record a complete reservation. This main program calls the `CheckHotelAvailability` function and the `RegisterReservation` procedure. We check the hotel availability and register a reservation if the hotel is available; otherwise, it notifies about registration failure.

1. CheckHotelAvailability.fnc

CREATE OR REPLACE FUNCTION ...

SQL Output Statistics

```
CREATE OR REPLACE FUNCTION CheckHotelAvailability(
    p_hotel_id IN INTEGER,
    p_date IN DATE
) RETURN BOOLEAN IS
    v_count INTEGER;
BEGIN
    SELECT COUNT(*) INTO v_count
    FROM RESERVATION
    WHERE HOTELID = p_hotel_id
      AND CHECKINDATE = p_date;

    IF v_count = 0 THEN
        RETURN TRUE;
    ELSE
        RETURN FALSE;
    END IF;
END CheckHotelAvailability;
/
```

2. RegisterReservation.prc

```
CREATE OR REPLACE FUNCTION ...  CREATE OR REPLACE PROCEDUR ...
SQL  Output  Statistics
CREATE OR REPLACE PROCEDURE RegisterReservation(
    p_reservation_id IN INTEGER,
    p_client_id IN INTEGER,
    p_hotel_id IN INTEGER,
    p_checkin_date IN DATE,
    p_checkout_date IN DATE
) AS
BEGIN
    INSERT INTO RESERVATION (RESERVATIONID, CLIENTID, HOTELID, CHECKINDATE, CHECKOUTDATE)
    VALUES (p_reservation_id, p_client_id, p_hotel_id, p_checkin_date, p_checkout_date);

    COMMIT;
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        RAISE;
END RegisterReservation;
/
```

3. MainProgram1.sql

```
SQL  Output  Statistics
BEGIN
    IF CheckHotelAvailability(1, TO_DATE('2023-07-01', 'YYYY-MM-DD')) THEN
        RegisterReservation(101, 201, 1, TO_DATE('2023-07-01', 'YYYY-MM-DD'), TO_DATE('2023-07-05', 'YYYY-MM-DD'));
    ELSE
        DBMS_OUTPUT.PUT_LINE('Hotel is not available for the given date.');
    END IF;
END;
/
```

before:

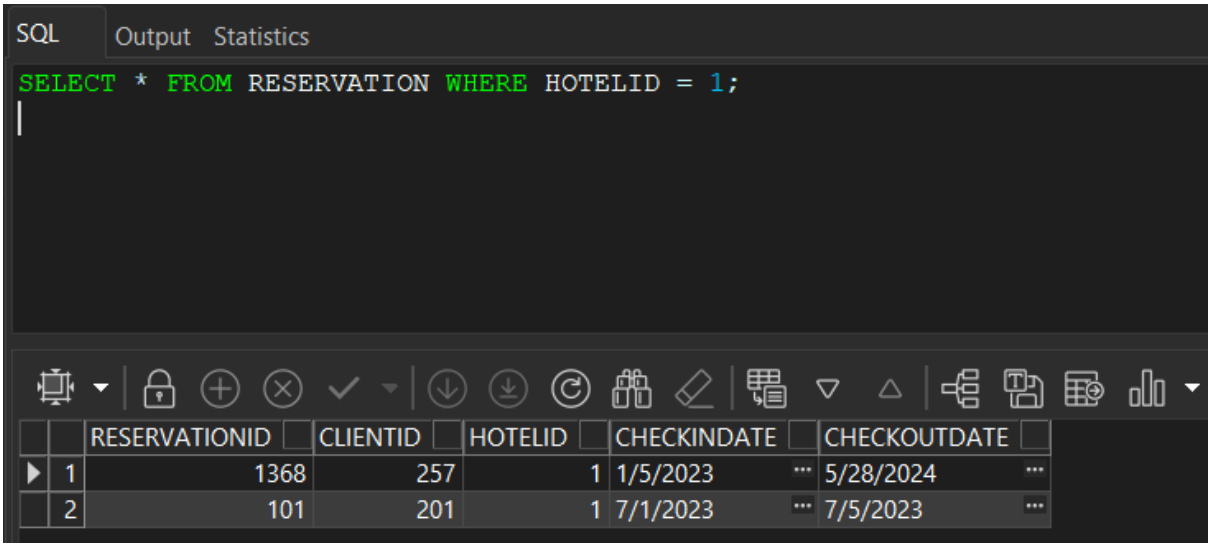
```
SQL  Output  Statistics
SELECT * FROM HOTEL WHERE HOTELID = 1;

SELECT * FROM RESERVATION WHERE HOTELID = 1;
```

Select hotel | Select reservation

| | RESERVATIONID | CLIENTID | HOTELID | CHECKINDATE | CHECKOUTDATE |
|---|---|---|---|---|---|
| 1 | 1368 | 257 | 1 | 1/5/2023 | 5/28/2024 |
| 2 | 101 | 201 | 1 | 7/1/2023 | 7/5/2023 |

after:

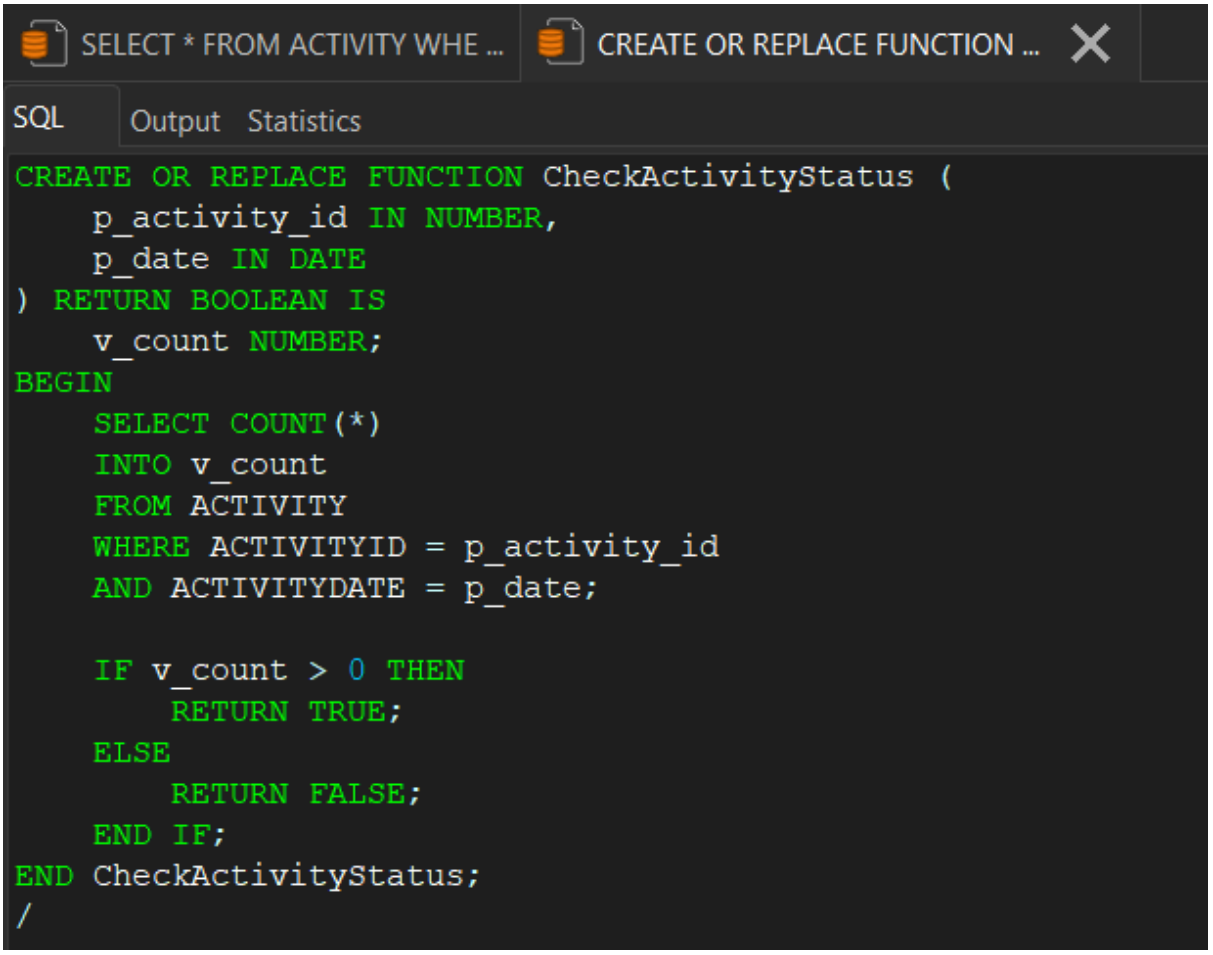



| | RESERVATIONID | CLIENTID | HOTELID | CHECKINDATE | CHECKOUTDATE |
|---|---|---|---|---|---|
| 1 | 1368 | 257 | 1 | 1/5/2023 | 5/28/2024 |
| 2 | 101 | 201 | 1 | 7/1/2023 | 7/5/2023 |

## Main2 - Program Description

Calculate booking days and update hotel availability. This main program calls the `CalculateTotalBookingDays` function and the `UpdateHotelAvailability` procedure. We calculate the total number of days booked for a client and update the hotel's availability for a given date.

1. Fonction `CheckActivityStatus`



Procédure `UpdateReservation`

```
SQL  Output  Statistics
CREATE OR REPLACE PROCEDURE UpdateReservation (
    p_reservation_id IN NUMBER,
    p_new_start_date IN DATE,
    p_new_end_date IN DATE
) IS
BEGIN
    UPDATE RESERVATION
    SET STARTDATE = p_new_start_date,
        ENDDATE = p_new_end_date
    WHERE RESERVATIONID = p_reservation_id;
END UpdateReservation;
/
```

3. Programme Principal `MainProgram2`

```
SQL  Output  Statistics
BEGIN
    IF CheckActivityStatus(1, TO_DATE('2023-08-01', 'YYYY-MM-DD')) THEN
        UpdateReservation(1000, TO_DATE('2023-08-01', 'YYYY-MM-DD'), TO_DATE('2023-08-05', 'YYYY-MM-DD'));
    ELSE
        DBMS_OUTPUT.PUT_LINE('Activity is not available for the given date.');
    END IF;
END;
/
```

before:

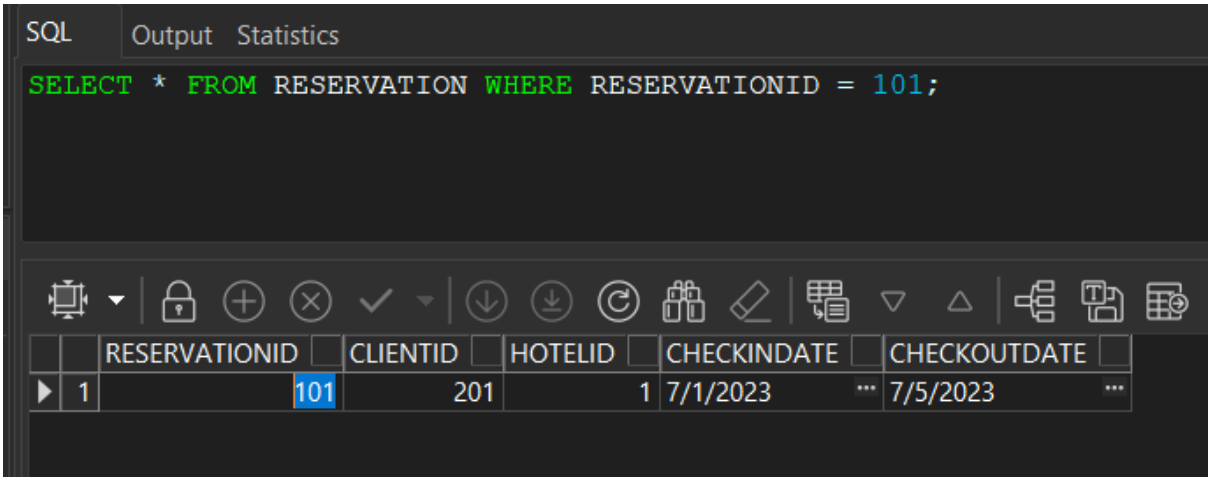
```
SQL  Output  Statistics
SELECT * FROM ACTIVITY WHERE ACTIVITYID = 1;

SELECT * FROM RESERVATION WHERE RESERVATIONID = 101;
```

Select activity | Select reservation

| | RESERVATIONID | CLIENTID | HOTELID | CHECKINDATE | CHECKOUTDATE |
|---|---|---|---|---|---|
| 1 | 101 | 201 | 1 | 7/1/2023 | 7/5/2023 |

after:


SQL
Output
Statistics
SELECT * FROM RESERVATION WHERE RESERVATIONID = 101;
RESERVATIONID
CLIENTID
HOTELID
CHECKINDATE
CHECKOUTDATE
1
101
201
1
7/1/2023
7/5/2023