



VIEWS IN SQL

--first row should always use this use database command

```
USE DATABASE DEMO_DATABASE;
```

--next create the required table

```
create or replace table aj_hospital_table (patient_id integer,
                                             patient_name varchar,
                                             billing_address varchar,
                                             diagnosis varchar,
                                             treatment varchar,
                                             cost number(10,2));
```

-- insert the data

```
insert into aj_hospital_table
  (patient_id, patient_name, billing_address, diagnosis, treatment, cost)
values
  (1, 'Mark Knopfler', '1982 Telegraph Road', 'Industrial Disease',
   'a week of peace and quiet', 2000.00),
  (2, 'Guido van Rossum', '37 Florida St.', 'python bite', 'anti-venom',
   70000.00),
  (3, 'Devin', '197 Brigade Road Texas', 'dog bite', 'Rabies Injection',
   40000.00),
  (4, 'Mark', '38 denver St Chicago', 'Dengue', 'Malaria',
   50000.00),
  (5, 'Peter', '78 New Yor City', 'Accident', 'Operation',
   340000.00);
```

--check the table and its contents

```
describe table aj_hospital_table ;
SELECT * FROM aj_hospital_table;
```



VIEWS IN SQL

--create view and --check its contents

```
create or replace view aj_doctor_view as
```

```
select patient_id, patient_name, diagnosis, treatment from aj_hospital_table;
```

```
describe view aj_doctor_view;
```

```
SELECT * FROM aj_DOCTOR_VIEW;
```

```
create or replace view aj_accountant_view as
```

```
select patient_id, patient_name, billing_address, cost from aj_hospital_table;
```

```
describe view aj_doctor_view;
```

```
SELECT * FROM aj_accountant_view;
```

-- A view can be used almost anywhere that a table can be used (joins, subqueries, etc.).

-- For example, using the views created above:

-- Show all of the types of medical problems for each patient:

```
select distinct diagnosis from aj_doctor_view;
```

-- A view can be used almost anywhere that a table can be used (joins, subqueries, etc.).

-- For example, using the views created above:

--Show all of the types of medical problems for each patient:

```
select distinct diagnosis from aj_doctor_view;
```

--Show the cost of each treatment (without showing personally identifying information about specific patients):

```
select treatment, cost
```

```
from aj_doctor_view as aj_dv, aj_accountant_view as aj_av
```



VIEWS IN SQL

```
where aj_av.patient_id = aj_dv.patient_id;
```

---A CREATE VIEW command can use a fully-qualified, partly-qualified, or unqualified table name.

--For example:

```
--create view v1 as select ... from my_database.my_schema.my_table;
```

```
create or replace view v1 AS
```

```
SELECT S_STORE_NAME,S_NUMBER_EMPLOYEES,S_HOURS
```

```
FROM "SNOWFLAKE_SAMPLE_DATA"."TPCDS_SF100TCL"."STORE";
```

```
SELECT * FROM V1 LIMIT 30;
```

--FROM RESPECTIVE SCHEMAS

```
create view v1 as select ... from my_schema.my_table;
```

--FROM RESPECTIVE TABLE

```
create view v1 as select ... from my_table;
```

--For example, you can create one view for the doctors, and one for the nurses,

-- and then create the medical_staff view by referring to the doctors view and nurses view:

```
create OR REPLACE table employees (id integer, title varchar);
```

```
insert into employees (id, title) values
```

```
    (1, 'doctor'),
```

```
    (2, 'nurse'),
```

```
    (3, 'janitor')
```

```
;
```

```
create view doctors as select * from employees where title = 'doctor';
```



VIEWS IN SQL

```
create view nurses as select * from employees where title = 'nurse';
```

```
create view medical_staff as
```

```
    select * from doctors
```

```
    union
```

```
    select * from nurses
```

```
;
```

```
select *
```

```
    from medical_staff
```

```
    order by id;
```

```
/*
```

Views Allow Granting Access to a Subset of a Table

Views allow you to grant access to just a portion of the data in a table(s).

For example, suppose that you have a table of medical patient records.

The medical staff should have access to all of the medical information (for example, diagnosis) but not the financial information

(for example, the patient's credit card number).

The accounting staff should have access to the billing-related information, such as the costs of each of the prescriptions given to the patient, but not to the private medical data, such as diagnosis of a mental health condition. You can create two separate views, one for the medical staff, and one for the billing staff, so that each of those roles sees only the information

needed to perform their jobs. Views allow this because you can grant privileges on a particular view to a particular role, without the grantee role having privileges on the table(s) underlying the view.

In the medical example:

The medical staff would not have privileges on the data table(s), but would have privileges on the view showing diagnosis and treatment.



VIEWS IN SQL

The accounting staff would not have privileges on the data table(s), but would have privileges on the view showing billing information.

Limitations on Views

The definition for a view cannot be updated

(i.e. you cannot use ALTER VIEW or ALTER MATERIALIZED VIEW to change the definition of a view).

To change a view definition, you must recreate the view with the new definition.

Changes to a table are not automatically propagated to views created on that table.

For example, if you drop a column in a table, the views on that table might become invalid.

Views are read-only (i.e. you cannot execute DML commands directly on a view).

However, you can use a view in a subquery within a DML statement that updates the underlying base table. For example:

```
*/  
delete from hospital_table  
where cost > (select avg(cost) from accountant_view);
```