**DB2 to Postgray sql with aws migration**

**Advantages of Postgre SQL over DB2**

1. Cost effectiveness: As PostgreSQL is an open source software and IBM DB2 is a closed source software, Postgre is cost saving.
2. Scalability: As the leading public cloud vendor AWS products is designed for security and scalability at a low cost. Scalability is enhanced with superior throughput and availability.
3. Flexibility: Postgres’ unique, multi-row approach to data storage also provides a responsive framework for high-demand deployments. In addition, storing json data in Postgres has been easy to do for several versions now and provides an excellent transition to or stand-in for “big data” operations.
4. Automated Backups: Turned on by default, the automated backup feature of Amazon RDS enables point-in-time recovery for your DB Instance. Amazon RDS will back up your database and transaction logs and store both for a user-specified retention period. This allows you to restore your DB Instance to any second during your retention period, up to the last five minutes. Your automatic backup retention period can be configured to up to thirty five days.
5. Fault recovery: The Startup and Hobbyist plans which do not have replica nodes are also automatically restored to their last good state without requiring user intervention should a failure occur. The perceived outage in such a situation depends on the database size
6. Migrate databases between clouds without downtime

**Create table**

CREATE TABLE "TestSM"."Employee"

(

emp\_id integer NOT NULL,

emp\_name character(20) COLLATE pg\_catalog."default" NOT NULL,

department integer NOT NULL,

address integer NOT NULL,

CONSTRAINT "Employee\_pkey" PRIMARY KEY (emp\_id),

CONSTRAINT fk\_add\_ky FOREIGN KEY (address)

REFERENCES "TestSM"."Address" (adr\_id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION,

CONSTRAINT fk\_dept\_id FOREIGN KEY (department)

REFERENCES "TestSM".dept (dept\_id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

**Alter query**

ALTER TABLE "TestSM"."Address"

ADD CONSTRAINT fk\_address\_id FOREIGN KEY (adr\_id)

REFERENCES "TestSM"."Employee" (address) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION;

Insert Query

INSERT INTO "TestSM".dept(

dept\_id, description)

VALUES (1, 'CM'),(2, 'IT'),(3, 'ET');

**Simple Functions**

CREATE OR REPLACE FUNCTION totalRecords ()

RETURNS integer AS $total$

declare

total integer;

BEGIN

SELECT count(\*) into total FROM "TestSM"."Employee";

RETURN total;

END;

$total$ LANGUAGE plpgsql;