

## Comparison of SQL/DDL (Data Definition Language) Statements Between Infx and psql

**Note: In the text below, "psql" is used as an abbreviation for PostgreSQL. It does not signify the utility for executing ad hoc queries.**

- Basic CREATE TABLE statements same as Infx
  - psql does not recognize "username." before tablename
  - psql does not recognize "{}" as comments - generates parse error AND CONTINUES (psql does recognize "--" as a comment as does Infx)
  - in psql, "CREATE TABLE tablename ( column1 ..." generates syntax error
    - extra space not allowed between "(" and column1
  
- ALTER TABLE tablename MODIFY columnname
  - available in psql but with different syntax
  - psql: ALTER TABLE tablename ALTER columnname
  - Infx allows the data type of the column to be changed using "MODIFY"
  - psql does not allow the data type to be changed using "MODIFY"
  
- ALTER TABLE location ADD CONSTRAINT PRIMARY KEY (lid) CONSTRAINT loc\_pk;
  - available in psql but with different syntax
  - Infx: ALTER TABLE location ADD CONSTRAINT PRIMARY KEY (lid) CONSTRAINT loc\_pk;
  - psql: ALTER TABLE location ADD CONSTRAINT loc\_pk PRIMARY KEY (lid);
  - can also name constraint from within CREATE TABLE statement as:
    - Infx: PRIMARY KEY (lid) CONSTRAINT location\_pk;
    - psql: CONSTRAINT location\_pk PRIMARY KEY (lid);
  - psql automatically defines columns in a primary key as NOT NULL
  - psql: max number of chars in constraint name = 63
  
- ALTER TABLE tablename ADD CONSTRAINT FOREIGN KEY ...
  - available in psql but with different syntax
  - Infx: ALTER TABLE location ADD CONSTRAINT FOREIGN KEY (tzone) REFERENCES TimeZone CONSTRAINT location\_tz\_fk;
  - psql: ALTER TABLE location ADD CONSTRAINT location\_tz\_fk FOREIGN KEY (tzone) REFERENCES TimeZone(tzone) MATCH FULL;
  - psql syntax needs additional field name which is not always part of Infx statement
    - in hd\_ob5xxx, constraints counties\_primbk\_fk and counties\_secbk\_fk have different column names in the tables being referenced (counties.primary\_back and counties.secondary\_back reference wfo.wfo)
  
- ALTER TABLE ... SET STATISTICS ...

- available in psql to determine "bucket size" for statistics on a column
- can influence scan method
  
- RENAME DATABASE oldname TO newname;
  - available in psql but with different syntax
  - psql: ALTER DATABASE oldname RENAME TO newname;
  - user must be owner of db AND have CREATEUSER privilege
  
- GRANT/REVOKE statements
  - psql and Infx have same statement syntax
  - database level privileges differ between Infx and psql
    - in psql, if user A creates a db, then user B automatically has access to it
      - Note: users A and B must be known to psql through the createuser command
    - in Infx: if user A creates a db, then user B does NOT have access to it unless granted CONNECT, RESOURCE or DBA privilege by user A
  - table level privileges also differ
    - in Infx, if user A creates a table, all users have all privileges on the table (unless they are revoked)
    - in psql, if user A creates a table, users have NO privileges on the table (unless they are granted) (see Douglas pp741-742)

### **Data Types in psql**

- DATE, TIME and INTERVAL data types available
- no DATETIME data type
  - TIMESTAMP data type for date/time exists
    - SELECT CURRENT\_TIMESTAMP generates current time
    - assumes yyymmddhhmmss
    - includes a time zone
    - psql data type TIMESTAMP WITHOUT TIME ZONE is equivalent to Infx data type DATETIME
    - data type TIMESTAMP defaults to TIMESTAMP WITHOUT TIME ZONE
    - all psql date/time types have precision of microseconds
- Infx DATETIME value = "08-01" is not allowed in psql – must specify year to create valid DATE field
  
- Infx FLOAT corresponds to psql FLOAT8
- Infx SMALLFLOAT corresponds to psql FLOAT4
- in psql, FLOAT is the same as FLOAT8
  
- Infx: SELECT ... WHERE ... column\_name = CURRENT
- psql: SELECT ... WHERE ... column\_name = CURRENT\_TIMESTAMP

### **Date/Time Functions**

Infx: select \* from precip where obstime > today;

- where today translates into hour = 0 date = today's date  
psql: select \* from precip where obstime > 'today';  
- today surrounded by single quotes

Infx: select \* from TextProduct where (CURRENT - postingtime) < 5 UNIT DAY;

psql: select \* from TextProduct  
where EXTRACT (DAY from CURRENT\_TIMESTAMP - postingtime) < 5;

Infx: select \* from ... where CURRENT - INTERVAL 5 DAY TO DAY;

psql: select \* from ... where CURRENT\_TIMESTAMP - INTERVAL 5 DAY;

Note: The above Infx syntax is not caught by the psql syntax checker. Instead, the following error message is generated in the postgres log:

ERROR: syntax error at or near "DAY" at character ...

Infx: select \* from height where DATE(obstime) = "2006-02-15";

Psq: there is no equivalent to the DATE function which "extracts" a date from a timestamp data value

To list all records in the Height table which were posted in the last 20 minutes:

SELECT \* FROM Height WHERE obstime > (now() - interval '20 min');

see Schonig pp 620 - 626 for info on TIMESTAMP, INTERVAL types and examples of manipulating TIMESTAMP values using INTERVAL values

In psql, to insert a record containing a NULL DATE value through an INSERT statement:

INSERT INTO tablename VALUES(...,NULL,...);

Infx/esqlc has functions dttoftasc and dttoasc

in psql/ecpg, recommend using dttoasc only - dttoftasc exists but does not work the same as in Infx

- psql has no BYTE data type but has a BYTEA data type

- psql has an OID (Object ID) which is similar to Informix's rowid

- every row in a postgres db has a unique OID

- in Oracle and Informix, rowid is the fastest possible way to get to a record

- according to Douglas p 102, this is NOT true in postgres

Infx SMALLINT - psql INT2

Infx SMALLFLOAT - psql FLOAT4

Infx DECIMAL - psql NUMERIC

- according to the psql documentation, numeric type is equivalent to decimal type
- histdata db uses DECIMAL type
- DECIMAL type also appears in dbgen

Column Naming

- column names must begin with a letter or underscore
- max of 64 char

## **OUTER JOIN**

Infx: OUTER xxx

psql: LEFT/RIGHT/FULL OUTER JOIN xxx

- psql uses ANSI standard format for OUTER join (Infx does not)
- 8 out of 16 views in hd\_ob6xxx contain keyword OUTER
- Example:

- Infx: CREATE VIEW locview (...  
SELECT ...  
FROM location x0, outer riverstat x1  
WHERE x0.lid = x1.lid ;

- psql: CREATE VIEW locview (...  
SELECT ...  
FROM location x0 left outer join riverstat x1  
ON x0.lid = x1.lid  
WHERE ... ;

- comma between tables removed
- "WHERE" replaced with "ON"

- result sets from FpInfo and LocView views (both with OUTER keyword) between Infx and psql were compared and found to be the same

Infx: SET LOCK MODE TO WAIT nn;

- where nn = number of seconds
- can be set on a "transaction-by-transaction" basis

psql: SET STATEMENT\_TIMEOUT TO nn;

- where nn = number of milliseconds
- server run-time configuration parameter
- value of 0 (default) turns off timer
- see Section 16.4.7.1 of PostgreSQL 7.4.7 Documentation

psql has upper limit of 1600 columns for a table

- dropping a column actually hides the column from the user but still counts towards the 1600 column limit
- adding and dropping a column 1600 times will begin to show errors concerning

the 1600 column limit

### **Cursors**

Infx has EXEC SQL CLOSE and EXEC SQL FREE statements to close a cursor and free its resources

psql has EXEC SQL CLOSE statement which closes the cursor and frees all resources related to the cursor

psql has no FREE statement

psql has no "WHERE CURRENT OF ..." clause for cursors (will have in future release)

Infx allows an open cursor to be reopened without executing a CLOSE first - in psql, this results in a 42P03 error - in psql, must close a cursor before reopening it

### **Checking Query Plans**

Infx: SET EXPLAIN ON;  
      SELECT .... ;  
      SET EXPLAIN OFF;

psql: EXPLAIN SELECT ... ;  
psql: EXPLAIN ANALYZE SELECT ...;

### **Character Strings**

- psql: character strings in SQL statements must be denoted by single quotes (')
- Infx: accepts both single and double quotes

Example:

psql: SELECT \* FROM height WHERE lid = 'ABCD1';  
Infx: SELECT \* FROM height WHERE lid = "ABCD1";

To select lids from the Location table which begin with lower case characters:

```
SELECT lid FROM Location WHERE lid SIMILAR TO '[a-z]%' ;
```

### **Other SQL syntax Differences**

Returning only the first 10 records of a SELECT:

Infx: SELECT FIRST 10 \* FROM ...;  
psql: SELECT \* FROM ... LIMIT 10;

Infx: SELECT UNIQUE lid FROM CurPP;

psql SELECT DISTINCT lid FROM CurPP;

Note that Infx recognizes DISTINCT as a synonym to UNIQUE

ROUND function in psql requires first parameter to be numeric type - Infx requires that

first parameter be double precision type

Example:

```
Infx: SELECT ... WHERE lid = 'BLUO2' AND ROUND(stage,2) > 3.53;
```

```
psql: SELECT ... WHERE lid = 'BLUO2' AND ROUND(cast (stage as numeric), 2) > 3.53 ;
```

```
psql : SELECT ... WHERE lid='BLUO2' AND ROUND(stage::numeric),2) > 3.53;
```

In psql, to list all Location table identifiers with all lower case characters:

```
SELECT lid FROM Location WHERE lid ~ ('[a-z]');
```

“~” is a POSIX regular expression operator similar to “LIKE” (See Section 9.6.3)

### **Temp tables**

psql allows the user to create a temp table with the same name as a real table - the temp table will "mask" the real table during the session - temp table is dropped at the end of the session - idea can be used for testing

### **Cascading Updates and Deletes**

as part of a CREATE TABLE statement, a column can be defined as "ON UPDATE CASCADE" - this will cause updates to "cascade" from parent table to child table - can also be set up for deletes - see Momjian pp161, 162

### **Locking Tables**

Infx has LOCK and UNLOCK statements

psql has a LOCK statement but has no UNLOCK statement - ending the transaction  
unlocks the table