No	概要	対象	ユーザ	ツール（あれば）	コマンド等	確認（確認コマンド等）	備考	本編関連箇所

1	移行元スキーマ作成	任意のマシン	任意のユーザ	JdbcRunner	$ export
					CLASSPATH=/usr/lib/oracle/11.2/client64/lib/ojdbc6.jar:jdbcrunner.jar
					展開したディレクトリ/jdbcrunner-1.2.jar
					
					$ java JR jdbcrunnerを展開したディレクトリ/scripts/tpcc_load.js
							tpcc_load.jsを以下の様に修正
							var jdbcUrl = "jdbc:oracle:thin:@localhost:1521:orcl";
							var jdbcUser = "scott";
							var jdbcPass = "tiger";
								–

2	移行元テーブルへの行数・サイズ確認	任意のマシン			1.テーブル毎の行数を確認 SQL> SELECT TABLE_NAME, NUM_ROWS FROM USER_TABLES; （省略） TABLE_NAME NUM_ROWS ---------- ---------- HISTORY 120000 ITEM 100000 STOCK 400000 ORDERS 120000 （省略）		PostgreSQLにデータ移行後、テーブルの行数を確認	1.6.1
3	既存データの抽出	OracleDatabaseサーバ	oracle		$ time sh extract.sh real 0m42.368s user 0m16.477s sys 0m7.026s ※real表示された時刻を実行時間として記録	$ ls -l /tmp/*.csv -rw-rw-r--. 1 oracle oracle 343713855 2月 26 20:17 2013 /tmp/customer.csv -rw-rw-r--. 1 oracle oracle 29600 2月 26 20:16 2013 /tmp/district.csv -rw-rw-r--. 1 oracle oracle 68186446 2月 26 20:16 2013 /tmp/history.csv -rw-rw-r--. 1 oracle oracle 13104447 2月 26 20:16 2013 /tmp/item.csv -rw-rw-r--. 1 oracle oracle 4752594 2月 26 20:17 2013 /tmp/new_orders.csv -rw-rw-r--. 1 oracle oracle 832525128 2月 26 20:19 2013 /tmp/order_line.csv -rw-rw-r--. 1 oracle oracle 63782721 2月 26 20:17 2013 /tmp/orders.csv -rw-rw-r--. 1 oracle oracle 656009831 2月 26 20:17 2013 /tmp/stock.csv -rw-rw-r--. 1 oracle oracle 2592 2月 26 20:16 2013 /tmp/warehouse.csv	1. DATA型をPostgreSQLに扱える文字表現にする （TO_CHAR関数を使用） 実行ファイルについては、シート「A_1_extract.sh」参照	2.3.1

4	既存データの抽出	OracleDatabaseサーバ	oracle		$ time ora2pg
					
					real 2m29.831s
					user 2m15.736s
					sys 0m1.338s
					$ java JR jdbcrunnerを展開したディレクトリ/scripts/tpcc_load.js
						$ ls -l output.sql
						-rw-r--r--. 1 oracle oracle 1135048112 2月 26 20:40 2013 output.sql
							設定ファイル「/etc/ora2pg/ora2pg.conf」については、
							シート「A_2_ora2pg.conf」参照
								2.3.1

5	移行先DBの作成	PostgreSQLサーバ	postgres		$ initdb -E UTF-8 --no-locale -D /home/postgres/data/
					
					$ pg_ctl start
					$ psql -U postgres
					
					postgres=# CREATE DATABASE tpcc WITH OWNER postgres
					CONNECTION LIMIT=-1;
					CREATE DATABASE
					
					Postgres=# ¥l
					（省略）
								3.3.3

6	OracleDatabaseのオブジェクト権限の確認	OracleDatabaseサーバ	oracle		SQL> select table_name,GRANTEE,PRIVILEGE from USER_TAB_PRIVS where owner='SCOTT'; no rows selected		本件では、jdbcrunnerにscott以外のユーザがいないため、何も表示されません。	–
7	OracleDatabaseの制約および索引の確認	OracleDatabaseサーバ	oracle		1.制約の確認 SQL>select table_name,constraint_name from user_constraints; TABLE_NAME CONSTRAINT_NAME ------------------------------ ------------------------------ NEW_ORDERS NEW_ORDERS_FK1 ORDER_LINE ORDER_LINE_FK1 （省略） 2. 索引の確認 SQL>select TABLE_NAME,INDEX_NAME from user_indexes; TABLE_NAME INDEX_NAME ------------------------------ ------------------------------ WAREHOUSE WAREHOUSE_PK STOCK STOCK_PK ORDER_LINE ORDER_LINE_PK ORDERS ORDERS_PK ORDERS ORDERS_IX1 NEW_ORDERS NEW_ORDERS_PK ITEM ITEM_PK EMP PK_EMP DISTRICT DISTRICT_PK DEPT PK_DEPT （省略）			1.7.1

8	PostgreSQLのユーザの作成	PostgreSQLサーバ	postgres		$ psql -U postgres
					postgres=# create role scott with login password 'tiger';
					$ psql -U postgres -d tpcc
					tpcc=# create schema AUTHORIZATION scott;
							必要であれば、Oracleのscottと権限を合わせる	1.5.1

9	PostgreSQL テーブル定義	任意のマシン	–	–	$ psql -U scott -f create_postgres_table.sql tpcc	$ psql -U scott -d tpcc tpcc=> ¥d List of relations Schema	Name	Type	Owner --------+------------+-------+-------- scott	customer	table	scott scott	district	table	scott scott	history	table	scott scott	item	table	scott scott	new_orders	table	scott scott	order_line	table	scott scott	orders	table	scott scott	stock	table	scott scott	test	table	scott scott	warehouse	table	scott (10 rows)	実行ファイルについては、シート 「A_3_create_postgres_table.sql」参照	1.4

10	データの投入(COPY)	PostgreSQLサーバ	postgres	COPY	$ psql -U postgres -d tpcc
					tpcc=# ¥timing
					Timing is on.
					
					tpcc=# copy scott.warehouse from '/tmp/warehouse.csv' CSV;
					COPY 4
					Time: 11.434 ms
					
					tpcc=# copy scott.district from '/tmp/district.csv' CSV;
					COPY 40
					Time: 12.495 ms
					
					tpcc=# copy scott.customer from '/tmp/customer.csv' CSV;
					COPY 120000
					Time: 3489.972 ms
					
					tpcc=# copy scott.history from '/tmp/history.csv' CSV;
					COPY 120000
					Time: 601.041 ms
					
					tpcc= # copy scott.item from '/tmp/item.csv' CSV;
					COPY 100000
					Time: 623.251 ms
					
					tpcc= # copy scott.stock from '/tmp/stock.csv' CSV;
					COPY 400000
					Time: 7502.149 ms
					
					tpcc=# copy scott.orders from '/tmp/orders.csv' CSV;
					COPY 120000
					Time: 601.322 ms
					
					tpcc=# copy scott.new_orders from '/tmp/new_orders.csv' CSV;
					COPY 36000
					Time: 68.779 ms
					
					tpcc= #copy scott.order_line from '/tmp/order_line.csv' with (NULL '1900/01/01',format CSV);
					COPY 1199845
					Time: 8006.671
								4.1

11	データの投入(Ora2Pg)	PostgreSQLサーバ	postgres	COPY （Ora2Pg抽出）	$ time psql -f /tmp/output.sql tpcc （省略） real 0m16.841s user 0m0.535s sys 0m0.323s			4.2

12	データの投入
	(pg_bulkload)
		PostgreSQLサーバ	postgres	pg_bulkload	$ psql -U postgres -d tpcc
					tpcc=# CREATE EXTENSION pg_bulkload ;
					CREATE EXTENSION
					
					$ time pg_bulkload /home/postgres/tpcc_warehouse.ctl -d tpcc
					（省略）
					4 Rows successfully loaded.
					0 Rows not loaded due to parse errors.
					（省略）
					
					real 0m0.157s
					user 0m0.003s
					sys 0m0.004s
					
					$ time pg_bulkload /home/postgres/tpcc_distrinct.ctl -d tpcc
					（省略）
					40 Rows successfully loaded.
					（省略）
					real 0m0.103s
					user 0m0.002s
					sys 0m0.002s
					
					$ time pg_bulkload /home/postgres/tpcc_customer.ctl -d tpcc
					（省略）
					120000 Rows successfully loaded.
					（省略）
					real 0m1.954s
					user 0m0.002s
					sys 0m0.003s
					
					$ time pg_bulkload /home/postgres/tpcc_history.ctl -d tpcc
					（省略）
					120000 Rows successfully loaded.
					（省略）
					
					real 0m0.581s
					user 0m0.001s
					sys 0m0.004
					
					$ time pg_bulkload /home/postgres/tpcc_item.ctl -d tpcc
					（省略）
					100000 Rows successfully loaded.
					（省略）
					
					real 0m0.596s
					user 0m0.002s
					sys 0m0.004s
					
					$ time pg_bulkload /home/postgres/tpcc_stock.ctl -d tpcc
					（省略）
					400000 Rows successfully loaded.
					（省略）
					
					real 0m3.853s
					user 0m0.002s
					sys 0m0.003s
					
					$ time pg_bulkload /home/postgres/tpcc_orders.ctl -d tpcc
					（省略）
					120000 Rows successfully loaded.
					（省略）
					
					real 0m0.540s
					user 0m0.001s
					sys 0m0.003s
					
					$ time pg_bulkload /home/postgres/tpcc_new_orders.ctl -d tpcc
					（省略）
					36000 Rows successfully loaded.
					（省略）
					real 0m0.158s
					user 0m0.001s
					sys 0m0.007s
					
					$ time pg_bulkload /home/postgres/tpcc_order_line.ctl -d tpcc
					（省略）
					1199845 Rows successfully loaded.
					（省略）
					real 0m5.468s
					user 0m0.001s
					sys 0m0.001s
						エラーや警告が表示されず、successfully loaded の行数がOracleの移行元表の行数と一致していればOK	制御ファイルについては、シート「A_4_pg_bulkload制御Iファイル」参照
							
							$PGDATA/pg_bulkload ディレクトリの存在を確認する。存在しない場合は、作成する。
								4.3

13	データ投入数の確認	PostgreSQLサーバ	postgres		$ psql -U postgres tpcc
					1.テーブルの表示
					tpcc= # select tablename from pg_tables where schemaname = 'scott' order by tablename;
					
					tablename
					---------- 	
					customer
					district
					history
					item
					new_orders
					order_line
					orders
					stock
					warehouse
					(9 rows)
					
					2.テーブルの行数確認
					tpcc=# select count(*) from scott.テーブル名;
					
					tpcc=# select count(*) from scott.warehouse;
					count
					------- 	
					4
					(1 row)
					（他のテーブルは省略）
						項番2と合っているか確認する		5.3

csv	PostgreSQL
	INDEXおよび制約の作成
		PostgreSQLサーバ	postgres		1.プライマリキーおよび索引の定義
					$ time psql -U scott -f /tmp/create_postgres_index.sql -d tpcc
					
					psql:/tmp/create_postgres_index.sql:2: NOTICE: ALTER TABLE / ADD PRIMARY KEY will create implicit index "warehouse_pk" for table "warehouse"
					ALTER TABLE
					（省略）
					real 0m5.661s
					user 0m0.001s
					sys 0m0.002s
					
					2.外部キーの定義
					$ time psql -U scott -f /tmp/create_postgres_foreingkey.sql -d tpcc
					ALTER TABLE
					ALTER TABLE
					（省略）
					real 0m7.887s
					user 0m0.002s
					sys 0m0.002s
						1.プライマリキーおよび索引の確認
						$ psql -d tpcc
						
						tpcc=# ¥di scott.*
						List of relations
						Schema | Name | Type | Owner | Table
						---------+---------------+-------+--------+-------------
						scott | customer_ix1 | index | scott | customer
						scott | customer_pk | index | scott | customer
						scott | district_pk | index | scott | district
						scott | item_pk | index | scott | item
						scott | new_orders_pk | index | scott | new_orders
						scott | order_line_pk | index | scott | order_line
						scott | orders_ix1 | index | scott | orders
						scott | orders_pk | index | scott | orders
						scott | stock_pk | index | scott | stock
						scott | warehouse_pk | index | scott | warehouse
						(10 rows)
						
						2.外部キーの確認
						tpcc=# ¥d+ scott.history
						（省略）
						Foreign-key constraints:
						"history_fk1" FOREIGN KEY (h_w_id, h_d_id) REFERENCES scott.district(d_w_id, d_id)
						"history_fk2" FOREIGN KEY (h_c_w_id, h_c_d_id, h_c_id) REFERENCES scott.customer(c_w_i
						d, c_d_id, c_id)
						Has OIDs: no
						（他のテーブルも同様に確認）
							実行ファイルについては、シート「A_5_create_postgres_index.sql」
							シート「A_6_create_postgres_foreign_key.sql」を参照
								5.2

15	PostgreSQLのオブジェクト権限の確認	PostgreSQLサーバ	postgres		$ psql -U scott tpcc tpcc=> # ¥z scott.* Access privileges Schema | Name | Type | Access privileges | Column access privileges ---------+-------------+-------+-------------------+-------------------------- scott | customer | table | | scott | district | table | | scott | history | table | | scott | item | table | | scott | new_orders | table | | scott | order_line | table | | scott | orders | table | | scott | stock | table | | scott | test | table | | scott | warehouse | table | | (10 rows)			5.3

16	PostgreSQL
	ANALYZA/VACCUM
		PostgreSQLサーバ	postgres		$ psql -U postgres tpcc
					
					tpcc=# ¥timing
					tpcc=# VACUUM (FULL,ANALYZE,VERBOSE);
					Time: 15120.301 ms
						–		5.5

17	移行後のデータベースサイズ取得	PostgreSQLサーバ	postgres		$ psql -U postgres tpcc
					
					tpcc=# select pg_relation_size('scott.warehouse');
					pg_relation_size
					------------------ 	
					8192
					
					tpcc=# select pg_relation_size('scott.district');
					pg_relation_size
					------------------ 	
					8192
					
					tpcc=# select pg_relation_size('scott.customer');
					pg_relation_size
					------------------ 	
					74153984
					
					tpcc=# select pg_relation_size('scott.history');
					pg_relation_size
					------------------ 	
					10616832
					
					tpcc=# select pg_relation_size('scott.stock');
					Pg_relation_size
					------------------ 	
					141893632
					
					tpcc=# select pg_relation_size('scott.item');
					pg_relation_size
					------------------ 	
					10641408
					
					tpcc=# select pg_relation_size('scott.orders');
					pg_relation_size
					------------------ 	
					8192000
					
					tpcc=# select pg_relation_size('scott.new_orders');
					pg_relation_size
					------------------ 	
					1597440
					
					tpcc=# select pg_relation_size('scott.order_line');
					pg_relation_size
					------------------ 	
					121348096
								5.6

18	アプリケーションテスト	任意のマシン	任意のユーザ		$ export CLASSPATH=jdbcrunner展開したディレクトリ/jdbcrunner-1.2.jar
					
					$ java JR jdbcrunnerを展開したディレクトリ/scripts/tpcc.js
							tpcc.jsを以下の様に修正
							var jdbcUrl = "jdbc:postgresql://localhost:5432/tpcc";
							var jdbcUser = "scott";
							var jdbcPass = "tiger";
								5.7

A_1_extract.sh

1. extract.sh

```
#/bin/sh

SQLPLUS=/home/oracle/app/oracle/product/11.2.0/dbhome_1/bin
ORAUSER=scott/tiger@orcl

#Oracleからのデータ抽出

#1.itemテーブル
$SQLPLUS/sqlplus -s $ORAUSER @/tmp/extract/extract_item.sql > /tmp/item.csv

#2.historyテーブル
$SQLPLUS/sqlplus -s $ORAUSER @/tmp/extract/extract_history.sql > /tmp/history.csv

#3.warehouseテーブル
$SQLPLUS/sqlplus -s $ORAUSER @/tmp/extract/extract_warehouse.sql > /tmp/warehouse.csv

#4.districtテーブル
$SQLPLUS/sqlplus -s $ORAUSER @/tmp/extract/extract_district.sql > /tmp/district.csv

#5.customerテーブル
$SQLPLUS/sqlplus -s $ORAUSER @/tmp/extract/extract_customer.sql > /tmp/customer.csv

#6.stockテーブル
$SQLPLUS/sqlplus -s $ORAUSER @/tmp/extract/extract_stock.sql > /tmp/stock.csv

#7.ordersテーブル
$SQLPLUS/sqlplus -s $ORAUSER @/tmp/extract/extract_orders.sql > /tmp/orders.csv

#8.new_ordersテーブル
$SQLPLUS/sqlplus -s $ORAUSER @/tmp/extract/extract_new_orders.sql > /tmp/new_orders.csv

#9.order_lineテーブル
$SQLPLUS/sqlplus -s $ORAUSER @/tmp/extract/extract_order_line.sql > /tmp/order_line.csv
```

2. extract_item.sql

```
set heading off
set feedback off
set echo off
set termout off
set linesize 1000
set pagesize 0
set trimspool on
set colsep ','
spool /tmp/item.csv
select  i_id,
i_im_id,
'"' || i_name ||'"',||
i_price,
'"' || i_data || '"'
from item;
spool off
exit
```

3. extract_history.sql

```
set heading off
set feedback off
set echo off
set termout off
set linesize 1000
set pagesize 0
set trimspool on
set colsep ','
spool /tmp/history.csv
select h_c_id,
h_c_d_id,
h_c_w_id,
h_d_id,
h_w_id,
TO_CHAR(h_date,'YYYY/MM/DD HH24:MI:SS '),
h_amount,
'"' || h_data || '"'
from history;
spool off
exit
```

4. extract_warehouse.sql

```
set heading off
set feedback off
set echo off
set termout off
set linesize 1000
set pagesize 0
set trimspool on
set colsep ','
spool /tmp/warehouse.csv
set heading off
set feedback off
set echo off
set termout off
set linesize 1000
set pagesize 0
set trimspool on
set colsep ','
spool /tmp/warehouse.csv
select w_id,
'"' || w_name ||'",'||
'"' || w_street_1 ||'",'||
'"' || w_street_2 ||'",'||
'"' || w_city  ||'",'||
'"' || w_state ||'",'||
'"' || w_zip ||'",'||
w_tax,
w_ytd
from warehouse;
spool off
exit
```

5. extract_district.sql

```
set heading off
set feedback off
set echo off
set termout off
set linesize 1000
set pagesize 0
set trimspool on
set colsep ','
spool /tmp/district.csv
select
d_id,
d_w_id,
'"' || d_name ||'",'||
'"' || d_street_1 ||'",'||
'"' || d_street_2 ||'",'||
'"' || d_city  ||'",'||
'"' || d_state ||'",'||
'"' || d_zip ||'",'||
d_tax,
d_ytd,
d_next_o_id
from district;
spool off
exit
```

6. extract_customer

```
set heading off
set feedback off
set echo off
set termout off
set linesize 1000
set pagesize 0
set trimspool on
set colsep ','
spool /tmp/customer.csv
select
c_id,
c_d_id,
c_w_id,
'"' || c_first ||'",'||
'"' || c_middle ||'",'||
'"' || c_last ||'",'||
'"' || c_street_1 ||'",'||
'"' || c_street_2 ||'",'||
'"' || c_city ||'",'||
'"' || c_state ||'",'||
'"' || c_zip ||'",'||
'"' || c_phone  ||'",'||
TO_CHAR(c_since,'YYYY/MM/DD HH24:MI:SS '),
'"' || c_credit ||'",'||
c_credit_lim,
c_discount,
c_balance,
c_ytd_payment,
c_payment_cnt,
c_delivery_cnt,
'"' ||c_data || '"'
from customer;
spool off
exit
```

7_extract_stock.sql

```
set heading off
set feedback off
set echo off
set termout off
set linesize 1000
set pagesize 0
set trimspool on
set colsep ','
spool /tmp/stock.csv
select
s_i_id,
s_w_id,
s_quantity,
'"' || s_dist_01 ||'",'||
'"' || s_dist_02 ||'",'||
'"' || s_dist_03 ||'",'||
'"' || s_dist_04 ||'",'||
'"' || s_dist_05 ||'",'||
'"' || s_dist_06 ||'",'||
'"' || s_dist_07 ||'",'||
'"' || s_dist_08 ||'",'||
'"' || s_dist_09 ||'",'||
'"' || s_dist_10 ||'",'||
s_ytd,
s_order_cnt,
s_remote_cnt,
'"' || s_data || '"'
from
stock;
spool off
exit
```

8_extract_orders.sql

```
set heading off
set feedback off
set echo off
set termout off
set linesize 1000
set pagesize 0
set trimspool on
set colsep ','
spool /tmp/orders.csv
select
o_id,
o_d_id,
o_w_id,
o_c_id,
TO_CHAR(o_entry_d,'YYYY/MM/DD HH24:MI:SS '),
nvl(o_carrier_id,'0'),
o_ol_cnt,
o_all_local
from
orders;
spool off
exit
```

9_extract_new_orders.sql

```
set heading off
set feedback off
set echo off
set termout off
set linesize 1000
set pagesize 0
set trimspool on
set colsep ','
spool /tmp/new_orders.csv
select
no_o_id,
no_d_id,
no_w_id
from
new_orders;
spool off
exit
```

10_extract_order_line.sql

```
set heading off
set feedback off
set echo off
set termout off
set linesize 1000
set pagesize 0
set trimspool on
set colsep ','
spool /tmp/order_line.csv
select
ol_o_id,
ol_d_id,
ol_w_id,
ol_number,
ol_i_id,
ol_supply_w_id,
''|| nvl(TO_CHAR(ol_delivery_d,'YYYY/MM/DD HH24:MI:SS'),'1900/01/01')||',||
ol_quantity,
ol_amount,
'"' || ol_dist_info|| '"'
from
order_line;
spool off
exit
```

A_2_Ora2Pg.conf

1. /etc/ora2pg/ora2pg.conf変更箇所

項番	設定名	設定値	意味
1	ORACLE_DSN	dbi:Oracle:host=localhost;sid=orcl	Oracle Database接続先
2	ORACLE_USER	scott	
3	ORACLE_PWD	tiger	
4	SHEMA	scott	対象スキーマのオーナ
5	TYPE	COPY	
6	ALLOW	item history warehouse district customer stock orders new_orders order_line ※JdbcRunner Tiny TPCCテーブル一式を設定	対象テーブル

A_3_Create_postgres_table.sql

1. create_postgres_table.sql

※以下のテーブル定義は、JdbcRunnerの「tpcc_load.js」より抜粋しています。

```
--itemテーブルの作成
CREATE TABLE item (
i_id INTEGER,
i_im_id INTEGER,
i_name VARCHAR(24),
i_price DECIMAL(5, 2),
i_data VARCHAR(50));
#1.itemテーブル
--historyテーブルの作成
CREATE TABLE history (
h_c_id INTEGER,
h_c_d_id INTEGER,
h_c_w_id INTEGER,
h_d_id INTEGER,
h_w_id INTEGER,
h_date TIMESTAMP,
h_amount DECIMAL(6, 2),
h_data VARCHAR(24));

--warehouseテーブルの作成
CREATE TABLE warehouse (
w_id INTEGER,
w_name VARCHAR(10),
w_street_1 VARCHAR(20),
w_street_2 VARCHAR(20),
w_city VARCHAR(20),
w_state CHAR(2),
w_zip CHAR(9),
w_tax DECIMAL(4, 4),
w_ytd DECIMAL(12, 2));

--districtテーブルの作成
CREATE TABLE district (
d_id INTEGER,
d_w_id INTEGER,
d_name VARCHAR(10),
d_street_1 VARCHAR(20),
d_street_2 VARCHAR(20),
d_city VARCHAR(20),
d_state CHAR(2),
d_zip CHAR(9),
d_tax DECIMAL(4, 4),
d_ytd DECIMAL(12, 2),
d_next_o_id INTEGER);

--customerテーブルの作成
CREATE TABLE customer (
c_id INTEGER,
c_d_id INTEGER,
c_w_id INTEGER,
c_first VARCHAR(16),
c_middle CHAR(2),
c_last VARCHAR(16),
c_street_1 VARCHAR(20),
c_street_2 VARCHAR(20),
c_city VARCHAR(20),
c_state CHAR(2),
c_zip CHAR(9),
c_phone CHAR(16),
c_since TIMESTAMP,
c_credit CHAR(2),
c_credit_lim DECIMAL(12, 2),
c_discount DECIMAL(4, 4),
c_balance DECIMAL(12, 2),
c_ytd_payment DECIMAL(12, 2),
c_payment_cnt DECIMAL(4, 0),
c_delivery_cnt DECIMAL(4, 0),
c_data VARCHAR(500));

--stockテーブルの作成
CREATE TABLE stock (
 s_i_id INTEGER,
 s_w_id INTEGER,
 s_quantity DECIMAL(4, 0),
 s_dist_01 CHAR(24),
 s_dist_02 CHAR(24),
 s_dist_03 CHAR(24),
 s_dist_04 CHAR(24),
 s_dist_05 CHAR(24),
 s_dist_06 CHAR(24),
 s_dist_07 CHAR(24),
 s_dist_08 CHAR(24),
 s_dist_09 CHAR(24),
 s_dist_10 CHAR(24),
 s_ytd DECIMAL(8, 0),
 s_order_cnt DECIMAL(4, 0),
 s_remote_cnt DECIMAL(4, 0),
 s_data VARCHAR(50));

--ordersテーブル作成
CREATE TABLE orders (
o_id INTEGER,
o_d_id INTEGER,
o_w_id INTEGER,
o_c_id INTEGER,
o_entry_d TIMESTAMP,
o_carrier_id INTEGER,
o_ol_cnt DECIMAL(2, 0),
o_all_local DECIMAL(1, 0));

--new_ordersテーブル作成
CREATE TABLE new_orders (
no_o_id INTEGER,
no_d_id INTEGER,
no_w_id INTEGER);

--order_lineテーブル作成
CREATE TABLE order_line (
ol_o_id INTEGER,
ol_d_id INTEGER,
ol_w_id INTEGER,
ol_number INTEGER,
ol_i_id INTEGER,
ol_supply_w_id INTEGER,
ol_delivery_d TIMESTAMP,
ol_quantity DECIMAL(2, 0),
ol_amount DECIMAL(6, 2),
ol_dist_info CHAR(24));
```

A_4_pg_bulkload制御ファイル

1. tpcc_warehouse.ctl

```
#
# sample_csv.ctl -- Control file to load CSV input data
#
#    Copyright (c) 2007-2011, NIPPON TELEGRAPH AND TELEPHONE CORPORATION
#
OUTPUT = scott.warehouse                    # [<schema_name>.]table_name
INPUT = /tmp/warehouse.csv  # Input data location (absolute path)
TYPE = CSV                          # Input file type
QUOTE = "¥""                         # Quoting character
ESCAPE = ¥                          # Escape character for Quoting
DELIMITER = ","                     # Delimiter
```

2.tpcc_distrinct.ctl

```
#
# sample_csv.ctl -- Control file to load CSV input data
#
#    Copyright (c) 2007-2011, NIPPON TELEGRAPH AND TELEPHONE CORPORATION
#
OUTPUT = scott.district                    # [<schema_name>.]table_name
INPUT = /tmp/district.csv  # Input data location (absolute path)
TYPE = CSV                          # Input file type
QUOTE = "¥""                         # Quoting character
ESCAPE = ¥                          # Escape character for Quoting
DELIMITER = ","                     # Delimiter
```

3.tpcc_customer.ctl

```
#
# sample_csv.ctl -- Control file to load CSV input data
#
#    Copyright (c) 2007-2011, NIPPON TELEGRAPH AND TELEPHONE CORPORATION
#
OUTPUT = scott.customer                    # [<schema_name>.]table_name
INPUT = /tmp/customer.csv  # Input data location (absolute path)
TYPE = CSV                          # Input file type
QUOTE = "¥""                         # Quoting character
ESCAPE = ¥                          # Escape character for Quoting
DELIMITER = ","                     # Delimiter
```

4.tpcc_history.ctl

```
#
# sample_csv.ctl -- Control file to load CSV input data
#
#    Copyright (c) 2007-2011, NIPPON TELEGRAPH AND TELEPHONE CORPORATION
#
OUTPUT = scott.history                    # [<schema_name>.]table_name
INPUT = /tmp/history.csv  # Input data location (absolute path)
TYPE = CSV                          # Input file type
QUOTE = "¥""                         # Quoting character
ESCAPE = ¥                          # Escape character for Quoting
DELIMITER = ","                     # Delimiter
```

5.tpcc_item.ctl

```
#
# sample_csv.ctl -- Control file to load CSV input data
#
#    Copyright (c) 2007-2011, NIPPON TELEGRAPH AND TELEPHONE CORPORATION
#
OUTPUT = scott.item                    # [<schema_name>.]table_name
INPUT = /tmp/item.csv  # Input data location (absolute path)
TYPE = CSV                          # Input file type
QUOTE = "¥""                         # Quoting character
ESCAPE = ¥                          # Escape character for Quoting
DELIMITER = ","                     # Delimiter
```

6.pcc_stock.ctl

```
#
# sample_csv.ctl -- Control file to load CSV input data
#
#    Copyright (c) 2007-2011, NIPPON TELEGRAPH AND TELEPHONE CORPORATION
#
OUTPUT = scott.stock                    # [<schema_name>.]table_name
INPUT = /tmp/stock.csv  # Input data location (absolute path)
TYPE = CSV                          # Input file type
QUOTE = "¥""                         # Quoting character
ESCAPE = ¥                          # Escape character for Quoting
DELIMITER = ","                     # Delimiter
```

7.tpcc_orders.ctl

```
#
# sample_csv.ctl -- Control file to load CSV input data
#
#    Copyright (c) 2007-2011, NIPPON TELEGRAPH AND TELEPHONE CORPORATION
#
OUTPUT = scott.orders                    # [<schema_name>.]table_name
INPUT = /tmp/orders.csv  # Input data location (absolute path)
TYPE = CSV                          # Input file type
QUOTE = "¥""                         # Quoting character
ESCAPE = ¥                          # Escape character for Quoting
DELIMITER = ","                     # Delimiter
```

8.tpcc_new_orders.ctl

```
#
# sample_csv.ctl -- Control file to load CSV input data
#
#    Copyright (c) 2007-2011, NIPPON TELEGRAPH AND TELEPHONE CORPORATION
#
OUTPUT = scott.new_orders                    # [<schema_name>.]table_name
INPUT = /tmp/new_orders.csv  # Input data location (absolute path)
TYPE = CSV                          # Input file type
QUOTE = "¥""                         # Quoting character
ESCAPE = ¥                          # Escape character for Quoting
DELIMITER = ","                     # Delimiter
```

9.tpcc_order_line.ctl

```
#
# sample_csv.ctl -- Control file to load CSV input data
#
#    Copyright (c) 2007-2011, NIPPON TELEGRAPH AND TELEPHONE CORPORATION
#
OUTPUT = scott.order_line                    # [<schema_name>.]table_name
INPUT = /tmp/order_line.csv  # Input data location (absolute path)
TYPE = CSV                          # Input file type
QUOTE = "¥""                         # Quoting character
ESCAPE = ¥                          # Escape character for Quoting
DELIMITER = ","                     # Delimiter
NULL="1900/01/01"
```

A_5_create_postgres_index.sql

1. create_postgres_index.sql

※以下のインデックス定義は、JdbcRunnerの「tpcc_load.js」より抜粋しています。

```
ALTER TABLE warehouse ADD CONSTRAINT warehouse_pk
      PRIMARY KEY (w_id);

ALTER TABLE district ADD CONSTRAINT district_pk
      PRIMARY KEY (d_w_id, d_id);

ALTER TABLE customer ADD CONSTRAINT customer_pk
      PRIMARY KEY (c_w_id, c_d_id, c_id);

ALTER TABLE item ADD CONSTRAINT item_pk
      PRIMARY KEY (i_id);

ALTER TABLE stock ADD CONSTRAINT stock_pk
      PRIMARY KEY (s_w_id, s_i_id);

ALTER TABLE orders ADD CONSTRAINT orders_pk
      PRIMARY KEY (o_w_id, o_d_id, o_id);

ALTER TABLE new_orders ADD CONSTRAINT new_orders_pk
      PRIMARY KEY (no_w_id, no_d_id, no_o_id);

ALTER TABLE order_line ADD CONSTRAINT order_line_pk
      PRIMARY KEY (ol_w_id, ol_d_id, ol_o_id, ol_number);
```

A_6_create_postgres_foreign_key.sql

1.create_postgres_foreign_key.sql

※以下の外部キー定義は、JdbcRunnerの「tpcc_load.js」より抜粋しています。

```
ALTER TABLE district ADD CONSTRAINT district_fk1 FOREIGN KEY (d_w_id) REFERENCES warehouse (w_id);

ALTER TABLE customer ADD CONSTRAINT customer_fk1 FOREIGN KEY (c_w_id, c_d_id) REFERENCES district (d_w_id, d_id);

ALTER TABLE history ADD CONSTRAINT history_fk1 FOREIGN KEY (h_w_id, h_d_id) REFERENCES district (d_w_id, d_id);

ALTER TABLE history ADD CONSTRAINT history_fk2 FOREIGN KEY (h_c_w_id, h_c_d_id, h_c_id) REFERENCES customer (c_w_id, c_d_id, c_id);

ALTER TABLE stock ADD CONSTRAINT stock_fk1 FOREIGN KEY (s_w_id) REFERENCES warehouse (w_id);

ALTER TABLE stock ADD CONSTRAINT stock_fk2 FOREIGN KEY (s_i_id) REFERENCES item (i_id);

ALTER TABLE orders ADD CONSTRAINT orders_fk1 FOREIGN KEY (o_w_id, o_d_id, o_c_id) REFERENCES customer (c_w_id, c_d_id, c_id);

ALTER TABLE new_orders ADD CONSTRAINT new_orders_fk1 FOREIGN KEY (no_w_id, no_d_id, no_o_id) REFERENCES orders (o_w_id, o_d_id, o_id);

ALTER TABLE order_line ADD CONSTRAINT order_line_fk1 FOREIGN KEY (ol_w_id, ol_d_id, ol_o_id) REFERENCES orders (o_w_id, o_d_id, o_id);

ALTER TABLE order_line ADD CONSTRAINT order_line_fk2 FOREIGN KEY (ol_supply_w_id, ol_i_id) REFERENCES stock (s_w_id, s_i_id);
```