Time Zones

From the Big Picture

To the Small Picture

By

Edward Kosciuszko

Kosware Inc.

Sequel@Optonline.net
TIME ZONES

3 Different Time Zones

Server

Server Time Zone

Database Time Zone

Oracle

Client

Client Time Zone

12/9/2007 Edward Kosciuszko
SeQueL@Optonline.net
TIME ZONES

Considerations

• Date literal or current time
• Time is relative to user
• Current time in EST is different from PST
• 5:00 AM EST and 2:00 AM PST are equal
• 2007-11-04 01:30AM – is this the time before or after daylight savings took affect?
TIME ZONES

And More Considerations

- Date indicates when entered relative to user?
- What to do when time zone algorithm changes?
- Replication?
**TIME ZONES**

*Database Time Zone*

CREATE DATABASE ... SET TIMEZONE='GMT'

- Default is OS time zone offset (which means database time zone setting can be different if created during DST or not.)

- Used to normalize TIMESTAMP WITH LOCAL TIME ZONE and TIMESTAMP WITH TIME ZONE values

**RECOMMENDATION**

Always set to explicit offset (e.g. –4:00)
TIME ZONES

Database Time Zone

ALTER DATABASE SET TIMEZONE='EST'

• Only possible if no TIMESTAMP WITH LOCAL TIME ZONE columns

• Requires DB restart to take affect.

• DBTIMEZONE function references database time zone
Session Time Zone

ALTER SESSION SET TIME_ZONE = local
• sets to O/S time zone

ALTER SESSION SET TIME_ZONE = DBTIMEZONE
• sets to database time zone

ALTER SESSION SET TIME_ZONE = '-05:00'
• sets to explicit offset from GMT

ALTER SESSION SET TIME_ZONE = ‘America/New_York’
• sets to named region
TIME ZONES

4 Different Date Types

• DATE
• TIMESTAMP
• TIMESTAMP WITH TIME ZONE
• TIMESTAMP WITH TIME ZONE
TIME ZONES

DATE

• Year to second but no time zone
• Great for historical dates or dates based on server only.
• 7 bytes

TIMESTAMP

• DATE w/ fractional seconds; no time zone
• argument = fractional digits (default=6 max=9)
• default format: NLS_TIMESTAMP_FORMAT
• 7 to 11 bytes storage
TIME ZONES

TIMESTAMP WITH TIME ZONE

- timestamp w/ time zone offset
- timestamp is normalized to UTC
- default format: NLS_TIMESTAMP_TZ_FORMAT
- cannot be part of primary key or unique index (Oracle considers 2 dates that map to same UTC time to be equal.)
- fixed 13 bytes
TIME ZONES

TIMESTAMP WITH LOCAL TIME ZONE

• time normalized to DBTIMEZONE
• originating time zone is not stored
• default format: NLS_TIMESTAMP_TZ_FORMAT
• 7 - 11 bytes depending on fractional seconds
TIME ZONES

TIMESTAMP WITH TIME ZONE

SELECT value FROM v$nlsp_parameters
WHERE parameter = 'NLS_TIMESTAMP_TZ_FORMAT'

Entering literal:

INSERT …..VALUES (….., '12-MAR-06 10:00:00AM',…)

Default time zone: session time zone
CREATE TABLE timestamp_w_tz
(sess_tz VARCHAR2(30), tz_time TIMESTAMP WITH TIME ZONE);

ALTER SESSION SET TIME_ZONE = 'America/New_York'
INSERT INTO timestamp_w_tz values (SESSIONTIMEZONE,
'12-MAR-06 10:00:00AM')

ALTER SESSION SET TIME_ZONE = 'US/Pacific'
INSERT INTO timestamp_w_tz values (SESSIONTIMEZONE,
'12-MAR-06 10:00:00AM')

SELECT SESSIONTIMEZONE, sess_tz, tz_time FROM timestamp_w_tz
**TIME ZONES**

**TIMESTAMP WITH TIME ZONE**
Default Display = Entered Date

```
ALTER SESSION SET TIME_ZONE = 'America/New_York'

INSERT INTO timestamp_w_tz values (SESSIONTIMEZONE,
   TO_TIMESTAMP_TZ ('20060312 10:00:00 America/New_York',
   'YYYYYMMDD HH24:MI:SS TZR'))

ALTER SESSION SET TIME_ZONE = 'US/Pacific'

INSERT INTO timestamp_w_tz values (SESSIONTIMEZONE,
   TO_TIMESTAMP_TZ ('20060312 10:00:00 America/New_York',
   'YYYYYMMDD HH24:MI:SS TZR'))

SELECT SESSIONTIMEZONE, sess_tz, tz_time FROM timestamp_w_tz
```

<table>
<thead>
<tr>
<th>SESSIONTIMEZONE</th>
<th>SESS_TZ</th>
<th>TZ_TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>US/Pacific</td>
<td>US/Pacific</td>
<td>12-MAR-06 10.00.00.0000000000 AM AMERICA/NEW_YORK</td>
</tr>
<tr>
<td>US/Pacific</td>
<td>America/New_York</td>
<td>12-MAR-06 10.00.00.0000000000 AM AMERICA/NEW_YORK</td>
</tr>
</tbody>
</table>

12/9/2007
Edward Kosciuszko
SeQueL@Optonline.net
TIME ZONES

TIMESTAMP WITH TIME ZONE
INTERNAL STORAGE FORMAT

INSERT INTO timestamp_w_tz values ('AMERICA/NEW_YORK',
   TO_TIMESTAMP_TZ ('20060312 10:00:00 America/New_York',
   'YYYYYMMDD HH24:MI:SS TZR'));

INSERT INTO timestamp_w_tz values ('US/PACIFIC',
   TO_TIMESTAMP_TZ ('20060312 10:00:00 US/Pacific',
   'YYYYYMMDD HH24:MI:SS TZR'))

INSERT INTO timestamp_w_tz values ('CANADA/EASTERN',
   TO_TIMESTAMP_TZ ('20060312 10:00:00 Canada/Eastern',
   'YYYYYMMDD HH24:MI:SS TZR'))

INSERT INTO timestamp_w_tz values ('-5:00',
   TO_TIMESTAMP_TZ ('20060312 10:00:00 -5:00',
   'YYYYYMMDD HH24:MI:SS TZH:TZM')

INSERT INTO timestamp_w_tz values ('-4:00',
   TO_TIMESTAMP_TZ ('20060312 11:00:00 -4:00',
   'YYYYYMMDD HH24:MI:SS TZH:TZM')

Time zone by explicit offset
TIME ZONES

TIMESTAMP WITH TIME ZONE
INTERNAL STORAGE FORMAT

‘12-MAR-06 11.00.00.000000000 AM +00:00’
is stored as
120,106,3,12,12,1,1,0,0,0,0,20,60

1st byte = century (excess 100 notation); e.g. 120-100 = 20
2nd byte = year in century (excess 100 notation) ; e.g. 106-100 = 06
3rd byte = month; e.g. 3 = March
4th byte = day; e.g. 12
5th byte = hour (excess 1 notation); e.g. 12-1 = 11
6th byte = minute (excess 1 notation); e.g. 1–1 = 0
7th byte = second (excess 1 notation); e.g. 1-1 = 0
8th – 11th for fractional seconds – recall 2 digits per byte
12th & 13th for time zone; e.g. 20,60 indicates +00:00
TIME ZONES

TIMESTAMP WITH TIME ZONE
INTERNAL STORAGE FORMAT

• All timestamps normalized relative to UTC (GMT)

• America/New_York and Canada/Eastern have same offset but ID stored differently in 12th & 13th byte.

• Each value is unique but cannot create unique index.

<table>
<thead>
<tr>
<th>SESS_TZ</th>
<th>TZ_TIME</th>
<th>DUMP(TZ_TIME)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMERICA/NEW_YORK</td>
<td>12-MAR-06 10:00:00:000000000 AM AMERICA/NEW_YORK</td>
<td>Typ=181 Len=13: 120,106,3,12,16,1,1,0,0,0,0,129,144</td>
</tr>
<tr>
<td>US/PACIFIC</td>
<td>12-MAR-06 10:00:00:000000000 AM US/PACIFIC</td>
<td>Typ=181 Len=13: 120,106,3,12,19,1,1,0,0,0,0,137,156</td>
</tr>
<tr>
<td>CANADA/EASTERN</td>
<td>12-MAR-06 10:00:00:000000000 AM CANADA/EASTERN</td>
<td>Typ=181 Len=13: 120,106,3,12,16,1,1,0,0,0,0,137,232</td>
</tr>
<tr>
<td>-5:00</td>
<td>12-MAR-06 10:00:00:000000000 AM -05:00</td>
<td>Typ=181 Len=13: 120,106,3,12,16,1,1,0,0,0,0,15,60</td>
</tr>
<tr>
<td>-4:00</td>
<td>12-MAR-06 11:00:00:000000000 AM -04:00</td>
<td>Typ=181 Len=13: 120,106,3,12,16,1,1,0,0,0,0,16,60</td>
</tr>
<tr>
<td>-0:00</td>
<td>12-MAR-06 11:00:00:000000000 AM +00:00</td>
<td>Typ=181 Len=13: 120,106,3,12,12,1,1,0,0,0,0,20,60</td>
</tr>
</tbody>
</table>

16 –1 = 15 which is the GMT hour
TIME ZONES

TIMESTAMP WITH LOCAL TIME ZONE
Default Display Relative to Session

CREATE TABLE timestamp_w_ltz
(sess_tz VARCHAR2(30), tz_time TIMESTAMP WITH LOCAL TIME ZONE);

ALTER SESSION SET TIME_ZONE = 'America/New_York'
INSERT INTO timestamp_w_ltz values (SESSIONTIMEZONE, '12-MAR-06 10:00:00AM')

ALTER SESSION SET TIME_ZONE = 'US/Pacific'
INSERT INTO timestamp_w_ltz values (SESSIONTIMEZONE, '12-MAR-06 10:00:00AM')

ALTER SESSION SET TIME_ZONE = 'GMT'
INSERT INTO timestamp_w_ltz values (SESSIONTIMEZONE, '12-MAR-06 10:00:00AM')

Same time relative to session
### TIME ZONES

**TIMESTAMP WITH LOCAL TIME ZONE**

Default Display Relative to Session

```sql
ALTER SESSION SET TIME_ZONE = 'America/New_York'
SELECT SESSIONTIMEZONE, sess_tz, tz_time FROM timestamp_w_ltz
```

![Table 1](image1)

```sql
ALTER SESSION SET TIME_ZONE = 'US/Pacific'
SELECT SESSIONTIMEZONE, sess_tz, tz_time FROM timestamp_w_ltz
```

![Table 2](image2)
TIME ZONES

TIMESTAMP WITH LOCAL TIME ZONE
INTERNAL STORAGE FORMAT

• All dates stored in DBTIMEZONE
• Originating time zone not stored

ALTER SESSION SET TIME_ZONE = 'America/New_York'
SELECT DBTIMEZONE, sess_tz, TZ_OFFSET(sess_tz), tz_time, dump(tz_time)
FROM timestamp_w_ltz

<table>
<thead>
<tr>
<th>DBTIMEZONE</th>
<th>SESS_TZ</th>
<th>TZ_OFFSET(SESS_TZ)</th>
<th>TZ_TIME</th>
<th>DUMP(TZ_TIME)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-05:00</td>
<td>America/New_York</td>
<td>-05:00</td>
<td>12-MAR-06 10:00.00.00000000000 AM</td>
<td>Typ=231 Len=7: 120,106,3,12,11,1,1</td>
</tr>
<tr>
<td>-05:00</td>
<td>US/Pacific</td>
<td>-08:00</td>
<td>12-MAR-06 01:00.00.00000000000 AM</td>
<td>Typ=231 Len=7: 120,106,3,12,14,1,1</td>
</tr>
<tr>
<td>-05:00</td>
<td>GMT</td>
<td>+00:00</td>
<td>12-MAR-06 05:00.00.00000000000 AM</td>
<td>Typ=231 Len=7: 120,106,3,12,6,1,1</td>
</tr>
</tbody>
</table>

11-1=10
TIME ZONES

Session Time

• SYSDATE is only the date taken directly from the server

• SYSTIMESTAMP is SYSDATE with time zone (return type is TIMESTAMP WITH TIME ZONE)

```
SELECT sysdate, dbtimezone, systimestamp
FROM dual
```

<table>
<thead>
<tr>
<th>TO_CHAR(SYSDATE,'DD-MON-YY...</th>
<th>DBTIMEZONE</th>
<th>SYSTIMESTAMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>26-nov-2007 09:38:06 PM</td>
<td>-05:00</td>
<td>26-NOV-07 09.38.06.046000000 PM -04:00</td>
</tr>
</tbody>
</table>

NOTE server time zone is –4:00

12/9/2007 Edward Kosciuszko
SeQueL@Optonline.net
Session Time

- LOCALTIMESTAMP is current date and time relative to session time zone
- Cast as TIMESTAMP (i.e. time zone is lost) !!!

```
ALTER SESSION SET TIME_ZONE = '-3:00'

SELECT dbtimezone, systimestamp, localtimestamp
FROM dual
```

<table>
<thead>
<tr>
<th>DBTIMEZONE</th>
<th>SYSTIMESTAMP</th>
<th>LOCALTIMESTAMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>-05:00</td>
<td>27-NOV-07 06.59.22.8590000000 AM</td>
<td>-05:00 27-NOV-07 08.59.22.8590000000 AM</td>
</tr>
</tbody>
</table>

NOTE server time zone is –5:00
Session Time

INSERT INTO timestamp_w_tz
VALUES (sessiontimezone, localtimestamp)

SELECT * FROM timestamp_w_tz

<table>
<thead>
<tr>
<th>SESS_TZ</th>
<th>TZ_TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>03:00</td>
<td>27-NOV-07 09:05:44.70300000 AM -03:00</td>
</tr>
</tbody>
</table>

INSERT INTO timestamp_w_ltz
VALUES (sessiontimezone, localtimestamp)

SELECT * FROM timestamp_w_tz

<table>
<thead>
<tr>
<th>SESS_TZ</th>
<th>TZ_TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>03:00</td>
<td>27-NOV-07 09:09:21.76500000 AM</td>
</tr>
</tbody>
</table>

Time zone picked up from session

Continue >
**TIME ZONES**

**Time Zone Conversion**

```
SELECT sess_tz, tz_time, tz_time AT TIME ZONE '+5:00'
FROM timestamp_w_tz
```

<table>
<thead>
<tr>
<th>SESS_TZ</th>
<th>TZ_TIME</th>
<th>TZ_TIMEATIMEZONE'5:00'</th>
</tr>
</thead>
<tbody>
<tr>
<td>-03:00</td>
<td>27-NOV-07 09.05.44.7030000000 AM -03:00</td>
<td>27-NOV-07 05.05.44.7030000000 PM +05:00</td>
</tr>
</tbody>
</table>

```
ALTER SESSION SET TIME_ZONE = '-4:00'
SELECT sess_tz, tz_time, tz_time AT LOCAL
FROM timestamp_w_tz
```

<table>
<thead>
<tr>
<th>SESS_TZ</th>
<th>DBTIMEZONE</th>
<th>TZ_TIME</th>
<th>TZ_TIMEATLOCAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>-3:00</td>
<td>-05:00</td>
<td>27-NOV-07 09.05.44.7030000000 AM -03:00</td>
<td>27-NOV-07 08.05.44.7030000000 AM -04:00</td>
</tr>
</tbody>
</table>

12/9/2007
Edward Kosciuszko
SeQueL@Optonline.net

Continue >
### Time Zone Conversion

```sql
SELECT sess_tz, tz_time, tz_time
FROM timestamp_w_tz
AT TIME ZONE 'US/Pacific'
```

<table>
<thead>
<tr>
<th>SESS_TZ</th>
<th>TZ_TIME</th>
<th>TZ_TIMEATTIMEZONE'US/PACIFIC'</th>
</tr>
</thead>
<tbody>
<tr>
<td>-03:00</td>
<td>28-NOV-07 08.58.14.593000000 AM -03:00</td>
<td>28-NOV-07 03.58.14.593000000 AM US/PACIFIC</td>
</tr>
</tbody>
</table>

... AT TIME ZONE DBTIMEZONE ...

... AT TIME ZONE SESSIONTIMEZONE ...
**TIME ZONES**

Time Zone Conversion

```sql
SELECT sess_tz, tz_time,
CAST(tz_time AS TIMESTAMP WITH LOCAL TIME ZONE)
FROM timestamp_w_tz
```

<table>
<thead>
<tr>
<th>SESS_TZ</th>
<th>TZ_TIME</th>
<th>CAST(TZ_TIME AS TIMESTAMP WITH LOCAL TIME ZONE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-03:00</td>
<td>28-NOV-07 08:58:14.593000000 AM -03:00</td>
<td>28-NOV-07 08:58:14.593000000 AM</td>
</tr>
</tbody>
</table>

CAST(… AS TIMESTAMP)
CAST(… AS TIMESTAMP WITH TIME ZONE)
CAST(… AS TIMESTAMP WITH LOCAL TIME ZONE)
**TIME ZONES**

Time Zone Conversion

Standard ⟷ Daylight Savings

```
ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MON-YYYY HH24:MI:SS'

SELECT NEW_TIME('29-OCT-2006 01:23:45', 'EDT', 'EST')
FROM DUAL;
```

![New Date and Time](New%20Date%20and%20Time.png)

29-OCT-2006 00:23:45
CURRENT_TIMESTAMP

CURRENT_DATE

- current datetime in session time zone but no time zone

CURRENT_TIMESTAMP

- CURRENT_DATE plus session time zone
- use instead of SYSDATE when recording client datetime in applications
TIME ZONES

Criteria Performance

CREATE TABLE sales_copy
(prod_id NUMBER(6,0), cust_id NUMBER,
time_tz TIMESTAMP WITH TIME ZONE,
time_ltz TIMESTAMP WITH LOCAL TIME ZONE,
channel_id CHAR(1 BYTE), promo_id NUMBER(6,0),
quantity_sold NUMBER(3,0), amount_sold NUMBER(10,2) )

ALTER SESSION SET TIME_ZONE = 'America/New_York'

INSERT INTO sales_copy
SELECT prod_id, cust_id,
    to_timestamp(to_char(time_id,'DD-MON-RR HH:MI:SS AM')) at local,
    to_timestamp(to_char(time_id,'DD-MON-RR HH:MI:SS AM')) at local,
    ... FROM sales

• Based on Oracle’s SH demo account table, SALES.
• Tests on 9i (-5:00, 16K blocks) – 10g (-00:00) 8K
When executing INSERT in SQL*Developer.

When executing
INSERT in SQL*Developer.

ORA-01878: specified field not found in datetime or interval

? Checking trace of login to SQL*Plus vs SQL*Developer:

• SQL*Plus sets TIME_ZONE = '-5:00'
• SQL*Developer sets TIME_ZONE = 'America/New_York'
• April 5 2:00:00AM – April 5 2:59:59AM does NOT exist

SELECT time_tz FROM sales_copy WHERE time_tz
BETWEEN to_timestamp_tz('05-APR-98 2:00:00 AM -5:00')
AND to_timestamp_tz('05-APR-98 2:59:59 AM -5:00')

<table>
<thead>
<tr>
<th>TIME_TZ</th>
</tr>
</thead>
<tbody>
<tr>
<td>05-APR-98 02.00.00.00000000 AM -05:00</td>
</tr>
<tr>
<td>05-APR-98 02.00.00.00000000 AM -05:00</td>
</tr>
</tbody>
</table>

Bad Data!
Problem is that original data contained only EST times

How do you convert only the rows involving the DST change?

1. Convert each date time to GMT
2. Convert step 1 to TIMESTAMP string
3. Append ‘+00:00’ to string and convert to TIMESTAMP WITH TIME ZONE
4. Add “AT LOCAL” to convert TIMESTAMP in step 3 to the local time zone, America/New_York
### Criteria Performance

```sql
SELECT to_timestamp_tz (
    to_char(new_time(
        to_date('02-apr-2006 2:59:59AM', 'DD-MON-YYYY HH:MI:SS AM'),
        'EST','GMT'),
        'YYYY-MM-DD HH:MI:SS AM')||' +00:00',
    'YYYY-MM-DD HH:MI:SSAM TZH:TZM') AT LOCAL
FROM dual
```

Hour added for DST

```sql
SELECT to_timestamp_tz (
    to_char(new_time(
        to_date('02-apr-2006 1:59:59AM', 'DD-MON-YYYY HH:MI:SS AM'),
        'EST','GMT'),
        'YYYY-MM-DD HH:MI:SS AM')||' +00:00',
    'YYYY-MM-DD HH:MI:SSAM TZH:TZM') AT LOCAL
FROM dual
```

Hour NOT added because EST

12/9/2007 Edward Kosciuszko
SeQueL@Optonline.net

Continue >
Generic DATE to TIME ZONE Conversion

---

```
INSERT INTO sales_copy
SELECT prod_id, cust_id,
  to_timestamp_tz (  
    to_char(new_time(time_id,'EST','GMT'), 'YYYY-MM-DD HH:MI:SS AM')||' +00:00',
    'YYYY-MM-DD HH:MI:SSAM TZH:TZM') AT LOCAL,
  to_timestamp_tz (  
    to_char(new_time(time_id,'EST','GMT'), 'YYYY-MM-DD HH:MI:SS AM')||' +00:00',
    'YYYY-MM-DD HH:MI:SSAM TZH:TZM') AT LOCAL, ...
FROM sales_no_part
```

- SQL Assumes ALL date times are EST
- If only portion are EST, use WHERE criteria to identify
- NEW_TIME returns DATE
Criteria Performance

ALTER SESSION SET TIME_ZONE = 'US/Pacific'
SELECT count(*) FROM sales_copy
WHERE time_tz > '06-JAN-98 12:00:00 AM'

<table>
<thead>
<tr>
<th>SQL_TEXT</th>
<th>EXECUTIONS</th>
<th>CPU_TIME</th>
<th>ELAPSED_TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECT count(*) FROM sales_copy WHERE time_tz &gt; '06-JAN-98 12:00:00 AM'</td>
<td>1</td>
<td>515625</td>
<td>539552</td>
</tr>
</tbody>
</table>

ALTER SESSION SET TIME_ZONE = 'US/Pacific'
SELECT count(*) FROM sales_copy
WHERE time_ltz > '06-JAN-98 12:00:00 AM'

<table>
<thead>
<tr>
<th>SQL_TEXT</th>
<th>EXECUTIONS</th>
<th>CPU_TIME</th>
<th>ELAPSED_TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECT count(*) FROM sales_copy WHERE time_ltz &gt; '06-JAN-98 12:00:00 AM'</td>
<td>1</td>
<td>250000</td>
<td>246115</td>
</tr>
</tbody>
</table>

Significant performance gain

52% less CPU

Continue >
Criteria Performance

Convert constant to same time zone as that stored?

```
ALTER SESSION SET TIME_ZONE = 'US/Pacific'
```

<table>
<thead>
<tr>
<th>Criteria</th>
<th>9i</th>
<th>10g</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy WHERE time_tz &gt; from_tz (TIMESTAMP '1998-01-06 00:00:00', 'GMT')</td>
<td>EXECUTIONS: 1 CPU_TIME: 515625 ELAPSED_TIME: 509568</td>
<td>EXECUTIONS: 1 CPU_TIME: 241497 ELAPSED_TIME: 244511</td>
</tr>
<tr>
<td>Copy WHERE time_tz &gt; from_tz (TIMESTAMP '1998-01-06 00:00:00', 'US/Pacific')</td>
<td>EXECUTIONS: 1 CPU_TIME: 593750 ELAPSED_TIME: 583497</td>
<td>EXECUTIONS: 1 CPU_TIME: 336833 ELAPSED_TIME: 336960</td>
</tr>
<tr>
<td>Copy WHERE time_tz &gt; from_tz (TIMESTAMP '1998-01-06 00:00:00', 'America/New_York')</td>
<td>EXECUTIONS: 1 CPU_TIME: 421875 ELAPSED_TIME: 421973</td>
<td>EXECUTIONS: 1 CPU_TIME: 238975 ELAPSED_TIME: 238975</td>
</tr>
<tr>
<td>Copy WHERE time_tz &gt; '06-JAN-98 12:00:00 AM'</td>
<td>EXECUTIONS: 1 CPU_TIME: 515625 ELAPSED_TIME: 518099</td>
<td>EXECUTIONS: 1 CPU_TIME: 237224 ELAPSED_TIME: 237224</td>
</tr>
</tbody>
</table>

10g – Explicitly convert to time zone – 29% gain

12/9/2007 Edward Kosciuszko SeQueL@Optonline.net

Continue >
### TIME ZONES

**Criteria Performance**

<table>
<thead>
<tr>
<th>Condition</th>
<th>Executions</th>
<th>CPU Time</th>
<th>Elapsed Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>WHERE timeLtz = '06-JAN-98 12:00:00 AM'</td>
<td>1</td>
<td>265625</td>
<td>257240</td>
</tr>
<tr>
<td>WHERE timeLtz &gt; from_tz (TIMESTAMP '1998-01-06 00:00:00', 'GMT')</td>
<td>1</td>
<td>312500</td>
<td>299739</td>
</tr>
<tr>
<td>WHERE timeLtz &gt; from_tz (TIMESTAMP '1998-01-06 00:00:00', 'America/New_York')</td>
<td>1</td>
<td>234375</td>
<td>229383</td>
</tr>
<tr>
<td>WHERE timeLtz &gt; from_tz(timestamp '1998-01-06 00:00:00', 'US/Pacific')</td>
<td>1</td>
<td>312500</td>
<td>317272</td>
</tr>
</tbody>
</table>

**9i**

<table>
<thead>
<tr>
<th>Condition</th>
<th>Executions</th>
<th>CPU Time</th>
<th>Elapsed Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>where timeLtz &gt; from_tz(timestamp '1998-01-06 00:00:00', 'GMT')</td>
<td>1</td>
<td>6467805</td>
<td>6471821</td>
</tr>
<tr>
<td>where timeLtz &gt; from_tz(timestamp '1998-01-06 00:00:00', 'US/Pacific')</td>
<td>1</td>
<td>6469163</td>
<td>6489110</td>
</tr>
<tr>
<td>where timeLtz = '06-JAN-98 12:00:00 AM'</td>
<td>1</td>
<td>139098</td>
<td>167518</td>
</tr>
<tr>
<td>where timeLtz &gt; from_tz(timestamp '1998-01-06 00:00:00', 'America/New_York')</td>
<td>1</td>
<td>6462113</td>
<td>6465838</td>
</tr>
</tbody>
</table>

**10g**: DATE is 64 % faster than TIMESTAMP WITH LOCAL TIME ZONE!

---

12/9/2007

Edward Kosciuszko
SeQueL@Optonline.net

Continue >
**TIME ZONES**

 Indexed Criteria Performance

<table>
<thead>
<tr>
<th>EXECUTIONS</th>
<th>CPU_TIME</th>
<th>ELAPSED_TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9607</td>
<td>9607</td>
</tr>
<tr>
<td>1</td>
<td>1342</td>
<td>1342</td>
</tr>
<tr>
<td>1</td>
<td>1313</td>
<td>1313</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EXECUTIONS</th>
<th>CPU_TIME</th>
<th>ELAPSED_TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10818</td>
<td>10818</td>
</tr>
<tr>
<td>1</td>
<td>8729</td>
<td>8729</td>
</tr>
<tr>
<td>2</td>
<td>25520</td>
<td>25520</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EXECUTIONS</th>
<th>CPU_TIME</th>
<th>ELAPSED_TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>157379</td>
<td>165408</td>
</tr>
<tr>
<td>1</td>
<td>171972</td>
<td>171972</td>
</tr>
<tr>
<td>1</td>
<td>135648</td>
<td>142393</td>
</tr>
</tbody>
</table>

**TIMESTAMP** index performs better when fewer rows.

12/9/2007 Edward Kosciuszko SeQueL@Optonline.net
EXIT OLMONLINE.NET

TIME ZONES

TIMESTAMP WITH TIME ZONE - INDEX

CREATE INDEX i_tz ON sales_copy (time_tz)

• generates the following function-based index:

CREATE INDEX "SH"."I_TZ" ON "SH"."SALES_COPY" (SYS_EXTRACT_UTC("TIME_TZ"))

• SYS_EXTRACT_UTC converts to UTC timestamp

• Query does not require SYS_EXTRACT_UTC to gain access to index.
WHERE current_timestamp - time_ltz > 100

ORA-00932: inconsistent datatypes: expected INTERVAL got NUMBER

WHERE current_timestamp - time_ltz > INTERVAL '100 0:0:0' DAY(3) TO SECOND

• Difference of 2 timestamps is INTERVAL
• DAY(3) required since default digits for days is 2
**TIME ZONES**

Timestamp Arithmetic

WHERE current_timestamp - time_ltz > INTERVAL '100 0:0:0' DAY(3) TO SECOND

WHERE current_timestamp - INTERVAL '100 0:0:0' DAY(3) TO SECOND > time_ltz

<table>
<thead>
<tr>
<th>EXECUTIONS</th>
<th>CPU_TIME</th>
<th>ELAPSED_TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>WHERE current_timestamp - time_ltz &gt; INTERVAL '100 0:0:0' DAY(3) TO SECOND</td>
<td>1</td>
<td>23093750</td>
</tr>
<tr>
<td>WHERE current_timestamp - INTERVAL '100 0:0:0' DAY(3) TO SECOND &gt; time_ltz</td>
<td>1</td>
<td>33562500</td>
</tr>
</tbody>
</table>

9i

10g

• 9i results make no sense

• “current_timestamp – INTERVAL” makes more sense since performed only once for all rows evaluated

Edward Kosciuszko
SeQueL@Optonline.net

12/9/2007

Continue >
TIME ZONES

Timestamp Arithmetic

<table>
<thead>
<tr>
<th>Execution</th>
<th>CPU Time</th>
<th>Elapsed Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>16681053</td>
<td>16694403</td>
</tr>
<tr>
<td>1</td>
<td>17137657</td>
<td>17191052</td>
</tr>
<tr>
<td>1</td>
<td>11360730</td>
<td>11383211</td>
</tr>
<tr>
<td>1</td>
<td>8402531</td>
<td>8428084</td>
</tr>
</tbody>
</table>

- 10g test
- TIMESTAMP WITH TIME ZONE > 2x’s faster

WHERE CAST(current_timestamp - INTERVAL '100 0:0:0' DAY(3) TO SECOND AS TIMESTAMP WITH LOCAL TIME ZONE) > time_tz

2.25 x’s slower !?!?

Edward Kosciuszko
SeQueL@Optonline.net

Continue >
**TIME ZONES**

**Timestamp Arithmetic**

WHERE CAST(current_timestamp - INTERVAL '100 0:0:0' DAY(3) TO SECOND AS TIMESTAMP WITH LOCAL TIME ZONE) > time_tz

*With indexes* TIMESTAMP WITH TIME ZONE is 40% faster than DATE

12/9/2007

Edward Kosciuszko
SeQueL@Optonline.net

Continue >
TIME ZONES

Daylight Savings Time

2006 Daylight Savings Time Schedule
Start: (+1 hour) April 2, 2006 2:00:00 AM
End: (-1 hour) October 29, 2006 2:00:00 AM

On Apr 2nd, 2:00:00 – 2:59:59 AM does not exist
On Oct 29th, 1:00:00 – 1:59:59 AM is repeated

Boundary Problem
How do you distinguish the repeated times on Oct 29th?
Daylight Savings Time

Apr 2\textsuperscript{nd}, 2:00:00 – 2:59:59 AM does not exist

\[
\text{SELECT} \ \text{to\_char(to\_date('02-APR-2006 2:00:01 AM',}
\text{'DD-MON-YYYY HH:MI:SS AM'), 'DD-MON-YYYY HH:MI:SS AM')}\
\text{FROM dual}
\]

Time doesn’t exist!

\[
\text{SELECT to\_timestamp\_tz('02-APR-2006 1:59:59 AM America/New\_York',}
\text{'DD-MON-YYYY HH:MI:SS AM TZR')}\
\text{FROM dual}
\]
**TIME ZONES**

**Daylight Savings Time**

Apr 2\textsuperscript{nd}, 2:00:00 – 2:59:59 AM does not exist

```
SELECT to_timestamp_tz('02-APR-2006 1:59:59 AM America/New_York', 'DD-MON-YYYY HH:MI:SS AM TZR') + INTERVAL '0 0:0:1' DAY TO SECOND FROM dual
```

Advances hour due to DST

```
TO_TIMESTAMP_TZ('02-APR-20061:59:59AMAMERICA/NEWYORK', '02-APR-06 03.00.00.000000000 AM AMERICA/NEW_YORK')
```

```
SELECT to_timestamp_tz('02-APR-2006 2:00:00 AM America/New_York', 'DD-MON-YYYY HH:MI:SS AM TZR') FROM dual
```

ORA-01878: specified field not found in datetime or interval

---

12/9/2007 Edward Kosciuszko SeQueL@Optonline.net

Continue >
TIME ZONES

Daylight Savings Time

Oct 29th, 1:00:00 – 1:59:59 AM is repeated

SELECT to_timestamp_tz('29-OCT-2006 1:30:00 AM America/New_York', 'DD-MON-YYYY HH:MI:SS AM TZR TZD') FROM dual

ORA-01883: overlap was disabled during a region transition

ALTER SESSION SET ERROR_ON_OVERLAP_TIME = FALSE

Now “overlap” times are considered to be standard time (in this case EST).

Default: FALSE
Recommendation: Set TRUE to prohibit implicit entry.

Edward Kosciuszko
SeQueL@Optonline.net

Continue >
**TIME ZONES**

**Daylight Savings Time**

Oct 29th, 1:00:00 – 1:59:59 AM is repeated

```
SELECT to_char(to_timestamp_tz('29-OCT-2006 1:30:00 AM America/New_York',
  'DD-MON-YYYY HH:MI:SS AM TZR'),
  'DD-MON-YYYY HH:MI:SS AM TZR TZD')
FROM dual
```

How do you indicate time is EDT?
**TIME ZONES**

Daylight Savings Time

Forcing DST

```sql
SELECT to_timestamp_tz('29-OCT-2006 1:30:00 AM America/New_York EDT', 'DD-MON-YYYY HH:MI:SS AM TZR TZD') FROM dual
```

DST specified

```sql
TO_TIMESTAMP_TZ('29-OCT-20061:30:00AMAMERICA,NEW 29-OCT-06 01.30.00.00000000 AM AMERICA/NEW_YORK
```

```sql
SELECT to_timestamp_tz('29-OCT-2006 2:30:00 AM America/New_York EDT', 'DD-MON-YYYY HH:MI:SS AM TZR TZD') FROM dual
```

ORA-01857: not a valid time zone

DST ended at 1:59:59 EDT, so this is not a valid “EDT” time.

12/9/2007

Edward Kosciuszko

SeQueL@Optonline.net

Continue >
TIME ZONES

Daylight Savings Time

- DST information is limited in future and may change
- Explicitly specify DST for future dates

SELECT * FROM v$timezone_names WHERE tzname = 'America/New_York'

<table>
<thead>
<tr>
<th>TZNAME</th>
<th>TZABBREV</th>
</tr>
</thead>
<tbody>
<tr>
<td>America/New_York</td>
<td>LMT</td>
</tr>
<tr>
<td>America/New_York</td>
<td>EST</td>
</tr>
<tr>
<td>America/New_York</td>
<td>EDT</td>
</tr>
<tr>
<td>America/New_York</td>
<td>EWT</td>
</tr>
</tbody>
</table>
TIME ZONES

Daylight Savings Time

- Environmental variable, ORA_TZFILE, identifies Oracle time zone file.
- 2 files: in $ORACLE_HOME/oracore/zoneinfo
  - timezone.dat contains all time zones
  - timezone.dat contains common time zones
- Some clients contain time zone files (not thin JDBC client)
- Oracle JVM has time zone information embedded
- Java Runtime Environment (JRE) stores rules on DST
- Upgrade both ORACLE and OJVM for 2007
**TIME ZONES**

**Default Behavior**

```
SELECT to_timestamp(time_id),
       to_timestamp(to_char(time_id,'DD-MON-RR HH:MI:SS AM'))
FROM sales
```

<table>
<thead>
<tr>
<th>TO_TIMESTAMP(TIME_ID)</th>
<th>TO_TIMESTAMP(TO_CHAR(TIME_ID,'DD-MON-RR HH:MI:SS AM'))</th>
</tr>
</thead>
<tbody>
<tr>
<td>28-JAN-98 12.00.00.000000000 AM</td>
<td>28-JAN-98 02.00.00.000000000 PM</td>
</tr>
<tr>
<td>28-JAN-98 12.00.00.000000000 AM</td>
<td>28-JAN-98 08.00.00.000000000 AM</td>
</tr>
<tr>
<td>28-JAN-98 12.00.00.000000000 AM</td>
<td>28-JAN-98 07.00.00.000000000 AM</td>
</tr>
<tr>
<td>28-JAN-98 12.00.00.000000000 AM</td>
<td>28-JAN-98 12.00.00.000000000 PM</td>
</tr>
<tr>
<td>28-JAN-98 12.00.00.000000000 AM</td>
<td>28-JAN-98 10.00.00.000000000 PM</td>
</tr>
<tr>
<td>28-JAN-98 12.00.00.000000000 AM</td>
<td>28-JAN-98 03.00.00.000000000 PM</td>
</tr>
</tbody>
</table>

**DATE data type**

W/o TO_CHAR, date truncated before converting to timestamp

<table>
<thead>
<tr>
<th>NAME</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUM_ROWS</td>
<td>1016271</td>
</tr>
<tr>
<td>BLOCKS</td>
<td>3400</td>
</tr>
<tr>
<td>AVG_ROW_LEN</td>
<td>47</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUM_ROWS</td>
<td>918643</td>
</tr>
<tr>
<td>BLOCKS</td>
<td>6228</td>
</tr>
<tr>
<td>AVG_ROW_LEN</td>
<td>47</td>
</tr>
</tbody>
</table>

12/9/2007  Edward Kosciuszko
SeQueL@Optonline.net  Continue >
TIME ZONES

Default Behavior

SELECT systimestamp, systimestamp + 1 FROM dual

```
SYSTIMESTAMP     SYSTIMESTAMP+1
30-NOV-07 06.20.49.578000000 PM -05:00  01-DEC-07
```

SELECT TO_CHAR(systimestamp + 1, 'DD-MON-RR HH:MI:SS TZR')
FROM dual

ORA-01821: date format not recognized

Addition changed TIMESTAMP to DATE data type.

SELECT systimestamp + INTERVAL ‘1 0:0:0’ DAY TO SECOND FROM dual

```
SYSTIMESTAMP+INTERVAL'10:0:0'DAYTOSECOND
01-DEC-07 06.19.27.953000000 PM -05:00
```
TIME ZONES

Which Datatype?

TIMESTAMP WITH TIME ZONE

- can determine when the data was entered relative to client
- date arithmetic more efficient since already in UTC
- must force transformation to session time zone

TIMESTAMP WITH LOCAL TIME ZONE

- automatically adjusts for client time zone querying
- replication: DB’s must have same DBTIMEZONE
- date arithmetic forces conversion to UTC
Questions?