

- Oracle Instant Client
- perl-DBI
- perl-DBD-Oracle

- CERN SLC:
 - Enable CERN repository in `/etc/yum.repos.d/slcXX-cernonly.repo`
 - `# yum update`
 - `# yum install perl-DBD-Oracle oracle-instantclient-basic perl-DBI`

- Recommended to install also:
 - `#yum install oracle-instantclient-tnsnames.ora`

Perl: Linux generic Installation

- Download instantclient:
 - <http://www.oracle.com/technetwork/indexes/downloads/index.html>
 - instantclient-basic-linux-11.2.0.3.0.zip, instantclient-sdk-linux-11.2.0.3.0.zip, instantclient-sqlplus-linux-11.2.0.3.0.zip
 - # mkdir /usr/local/oracle ; cd /usr/local/oracle
 - # ls /tmp/mypath/instantclient*.zip | xargs -n1 unzip -x
 - # echo '/usr/local/oracle/instantclient_11_2' > /etc/ld.so.conf.d/oracle.conf
 - # ldconfig
- Install/compile perl bindings and dependencies:
 - #export ORACLE_HOME=/usr/local/oracle/instantclient_11_2
 - #perl -MCPAN -e 'install DBD::Oracle'
- Get TNSNAMES.ORA
 - <http://service-oracle-tnsnames.web.cern.ch/service-oracle-tnsnames/tnsnames.ora>
 - /afs/cern.ch/project/oracle/admin/tnsnames.ora
 - Copy file in /etc

DBI: Connection (tnsnames)

```
#!/usr/bin/perl
use strict;
use DBI;
use DBD:Oracle qw(:ora_types);
# $ENV{'TNS_ADMIN'}='/directory/containing_tnsnames.ora_file';
# $ENV{'ORACLE_HOME'}='/directory/containing_oracle_installation';
my $dbh=DBI->connect("dbi:Oracle:devdb11"
    , "USER" , "PASSWORD"
    , { #DBI options
        RaiseError=>1 #Exception on error, default 0
        , AutoCommit=>0 #updates/inserts commit automatically, default 1
        , PrintError=>0 #Shows warnings on errors, default 1
        , LongReadLen=>5242880, LongTruncOk=>0 #Read max 5M on CLOBs,...
    }
);

if(!$dbh) {
    die "Error: $DBI::errstr";
}
```

DBI: Connection (full string)

```
#!/usr/bin/perl
use strict;
use DBI;
my $dbh=DBI->connect(q{dbi:Oracle:
  (DESCRIPTION=
    (ADDRESS= (PROTOCOL=TCP) (HOST=dbsrvg3305.cern.ch) (PORT=10121) )
    (ENABLE=BROKEN)
    (CONNECT_DATA=(SID=DEVDB11))
  })
  ,"USER" ,"PASSWORD"
  ,{ #DBI options
    RaiseError=>1 #Exception on error, default 0
    ,AutoCommit=>0 #updates/inserts commit automatically,default 1
  }
);

if(!$dbh) {
  die "Error: $DBI::errstr";
}
```

...

#Named param binding and result as array

```
my $qh = $dbh->prepare("select table_name, status,num_rows from  
user_tables where table_name like :myparam");
```

```
$qh->bind_param(":myparam","N%");
```

```
$qh->execute();
```

```
while(my(@result)=$qh->fetchrow_array()) {
```

```
    print "$result[0], $result[1], $result[2]\n";
```

```
}
```

```
$dbh->disconnect();
```

...

#Just positional and fetch by hash

```
my $qh = $dbh->prepare("select table_name,status,num_rows from  
user_tables where table_name like ?");
```

```
$qh->execute("N%");
```

```
while(my $r=$qh->fetchrow_hashref()) {  
    print "$r->{'TABLE_NAME'},"  
        ,"$r->{'STATUS'},"  
        ,"$r->{'NUM_ROWS'}\n";  
}
```

```
$dbh->disconnect();
```

- Fetch bound to scalar references

```
my($table_name,$status,$num_rows);  
$qh->bind_columns(\$table_name,\$status,\$num_rows);  
while($qh->fetch()) {  
    print "$table_name, $status, $num_rows\n";  
}
```

- Binding to specific type:

```
$qh->bind_param(":myparam",23,{ora_type=>ORA_NUMBER});
```

- Commit and rollback (only if no autocommit)

```
$qh->commit();
```

```
$qh->rollback();
```

- Prepared statements can be bound and executed several times.
- Enable trace/debugging:

```
$dbh->trace(NUM);#Or ora_verbose param in dbi connection
```

- Procedures:

```
my $qh = $dbh->prepare(q{
    begin myprocedure(?,?); end;
});
$qh->execute("a","b");
```

- Functions:

```
my $qh = $dbh->prepare(q{
    begin
        :result := myfunc();
    end;
});
my($maxlen,$result); $maxlen=100;
$qh->bind_param_inout(":result",\ $result, $maxlen);
$qh->execute();
print $result;
```


- `perldoc DBI`
- `perldoc DBD::Oracle`
- Perl ORM -> `DBIx::Class`

Jose Carlos Luna IT/CS