Creating Indexes on Foreign Keys

*by Jeff Hunter, Sr. Database Administrator*

**Problem**

Creating foreign keys constraints on tables increases the integrity of your data by preventing rows from being inserted into detail (sometimes called child tables) table that do not have a matching row in a master (also called the parent table) table.

The following code creates two tables: "EMP" and "DEPT". Both tables declare a primary key and the table "EMP" declares a foreign key constraint between "EMP" and "DEPT".

CREATE TABLE dept (

deptno NUMBER(2) CONSTRAINT PK\_DEPT PRIMARY KEY

, dname VARCHAR2(14)

, loc VARCHAR2(13)

);

CREATE TABLE emp (

empno NUMBER(4) CONSTRAINT PK\_EMP PRIMARY KEY

, ename VARCHAR2(10)

, job VARCHAR2(9)

, mgr NUMBER(4)

, hiredate DATE

, sal NUMBER(7,2)

, comm NUMBER(7,2)

, deptno NUMBER(2)

);

ALTER TABLE emp

ADD CONSTRAINT emp\_fk1

FOREIGN KEY (deptno)

REFERENCES dept (deptno);

Once this constraint is enabled, attempting to insert an "EMP" record with an invalid DEPTNO, or trying to delete a DEPTNO row that has matching "EMP" records, will generate an error. However, in order to preserve integrity during the operation, Oracle needs to apply a full "table-level" lock (as opposed to the usual row-level locks) to the child table when the parent table is modified.

**Solution**

By creating an index on the foreign key of the child table, these "table-level" locks can be avoided. (for instance, creating a foreign key on "EMP.DEPTNO").

CREATE INDEX emp\_n1

ON emp(deptno)

TABLESPACE indx;

Keep in mind that you will often be creating an index on the foreign keys in order to optimize join and queries. However, if you fail to create such a foreign key index and if the parent table is subject to updates, you may see heavy lock contention. If ever in doubt, it's often safer to create indexes on ALL foreign keys, despite the possible overhead of maintaining unneeded indexes.