PROBLEEM:

De ruimte op de hard-disk E: op de ORACLEPROD-TEST is te klein om de wekelijkse FULL-BACKUP op te slaan, vervolgens alle ARCHIVE-LOGS van de komende week, en de nieuwe wekelijkse FULL-BACKUP.

Nu moet er vaak handmatig een FULL-BACK verwijderd worden om een nieuwe FULL-BACKUP mogelijk te maken.

Situatie:

Door het handmatig verwijderen van een FULL-BACKUP die eigenlijk nog nodig is voor een RESTORE zorgt ervoor dat deze backup volgens RMAN = EXPIRED wordt. Doordat we geen retention-period ingesteld hebben ruimt RMAN na het maken van een FULL-BACKUP de vorige full-backup pas op.

Doordat we zelf geen CROSSCHECK of DELETE-EXPIRED draaien weet RMAN niet dat de vorige full-backup handmatig verwijderd is.

Oplossing

DELETE EXPIRED does not delete any files at all.  
CROSSCHECK compares the record of backups, as recorded in the control file, with the files that are actually on disk. If a backup file recorded in the repository is found to no longer exist, its entry in the repository is marked 'expired'. Then, DELETE EXPIRED simply deletes those records from the repository. It does not delete any actual files because and "expired" file, by definition, does not exist to be deleted. There is no connection whatsoever between 'expired' and 'obsolete'.  
  
The only way a backup file becomes 'expired' is if you manually delete it with an OS command instead of letting rman handle it. And you shouldn't be doing that. As for "I dont want to keep any obsolete backups", that's what the rman 'delete obsolete' is for. That's why you'd run 'delete obsolete' as part of every backup script. If you are short on disk space, you need to change your retention policy, affecting when a backup becomes 'obsolete', or buy more disk. Disk is cheap. Especially when measured against the cost of not being able to recover your database because you thought you knew more than Oracle about when to delete a backup file.  
  
And if you are thinking about shortening your recovery window (say from 7 days to 4 days), you need to adjust the schedule of when you take your level 0 backups. If your recovery window goes back to a point prior to your most recent level 0, then the previous level 0 will never be obsolete, regardless of how old it is.

Huidige instelling:

CONFIGURE RETENTION POLICY TO REDUNDANCY 1

The REDUNDANCY parameter of the CONFIGURE RETENTION POLICY command specifies how many full or level 0 backups of each data file and control file that RMAN should keep. If the number of full or level 0 backups for a specific data file or control file exceeds the REDUNDANCY setting, then RMAN considers the extra backups as obsolete. The default retention policy is REDUNDANCY 1.

As you produce more backups, RMAN keeps track of which ones to retain and which are obsolete. RMAN retains all archived logs and incremental backups that are needed to recover the nonobsolete backups.

Assume that you make a full backup of data file 7 on Monday, Tuesday, Wednesday, and Thursday. You now have four full backups of this data file. If REDUNDANCY is 2, then the Monday and Tuesday backups are obsolete. If you make another backup on Friday, then the Wednesday backup of data file 7 becomes obsolete.

Assume a different case in which REDUNDANCY is 1. You run a level 0 database backup at noon on Monday, a level 1 cumulative backup at noon on Tuesday and Wednesday, and a level 0 backup at noon on Thursday. Immediately after each daily backup you run the command DELETE OBSOLETE. The Wednesday DELETE command does not remove the Tuesday level 1 backup because this backup is not redundant: the Tuesday level 1 backup could be used to recover the Monday level 0 backup to a time between noon on Tuesday and noon on Wednesday. However, the DELETE command on Thursday removes the previous level 0 and level 1 backups.