

### 5.6.3 Utenze

Valgono i principi generali già introdotti nel paragrafo [rif. 5.1.15.1.1].

### 5.6.4 Autenticazione

Valgono i principi generali già introdotti nel paragrafo [rif. 5.1.2].

### 5.6.5 Autorizzazione

A i principi generali già introdotti nel paragrafo [rif. 0], si aggiungono le seguenti indicazioni per il contesto specifico:

Autorizzazione	
<b>Minaccia</b>	Accesso non autorizzato alle informazioni.
<b>Contromisure</b>	<p>Utilizzare e configurare opportunamente i meccanismi di controllo di accesso alle risorse (tabelle, viste, procedure, ecc.) gestite dal DBMS (a titolo di esempio l'istruzione "grant" fornita da Oracle), fornendo a ciascun utente o utenza applicativa i minimi diritti effettivamente necessari al corretto funzionamento, secondo il principio del <i>least privilege</i>.</p> <p>Ad es., evitare l'accesso di un application server al DBMS con utenza di amministratore globale del database, anche quando un utente non privilegiato deve effettuare compiti di ordinaria operatività.</p>

### 5.6.6 Crittografia

Ai principi generali già introdotti nel paragrafo [rif. 5.1.4], si aggiungono le seguenti indicazioni per il contesto specifico:

Protezione delle informazioni riservate	
<b>Minaccia</b>	<ul style="list-style-type: none"> <li>- Crittografia debole o non validata.</li> <li>- Accesso non autorizzato alle informazioni.</li> </ul>
<b>Contromisure</b>	<ul style="list-style-type: none"> <li>- Per la protezione delle informazioni riservate custodite nel database, all'interno di campi specifici di tabelle specifiche, utilizzare tecniche di crittografia dei dati a livello di colonna fornite nativamente dallo specifico DBMS, evitando l'uso di soluzioni custom o di terze parti, evitando così tutta una serie di problemi che possono sorgere (ad es. quando la colonna cifrata è parte di un indice: in tal caso solitamente si indicizza il valore cifrato anziché quello in chiaro);</li> <li>- Per la protezione delle informazioni riservate custodite nel database, all'interno di righe specifiche di una tabella, utilizzare tecniche di crittografia dei dati a livello di riga fornite nativamente dallo specifico DBMS, evitando l'uso di soluzioni custom o di terze parti;</li> <li>- In presenza di dati particolarmente sensibili, valutare se vi sia davvero la necessità di custodirli all'interno del DBMS, e in caso contrario, evitare la loro memorizzazione permanente;</li> <li>- I dati sensibili presenti sui DBMS di produzione non devono mai essere trasferiti su sistemi di sviluppo, test e collaudo, se non dopo essere stati sottoposti ad un</li> </ul>